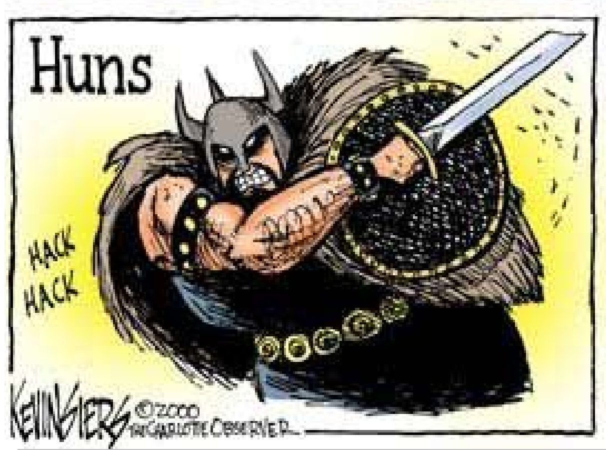


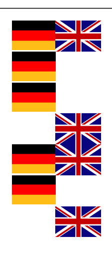
BRINGING CIVILIZATION TO ITS KNEES...











Herausgeber und für den Inhalt verantwortlicher:
Leszek Chmielewski, Prager Straße 92/11/12, 1210 Wien, Österreich
@Mail: retrozx@gmail.com
Klubkonto (Inhaber: Bernhard Lutz):
KTO.: 546143, BLZ: 54862500 (VR Bank Südpfalz, Sitz: Landau)
SWIFT-CodeGENODE6K, BIC-CodeGENODE61SUW

Ausgabe 224

1 Quartal 2011

Das Vorwort

http://www.womoteam.de/
http://spc.tlienhard.com/

Willkommen zu der Zeitschrift von Usern für User. Wir sind vor allem auf EURE Artikel angewiesen. Ich kann alleine keine (angepeilten) 24-32 Seiten füllen, so gerne ich es auch tun würde. Ehrenwort! Für eingeschickte Artikel gelten folgende Regeln:
Die Artikel müssen sich mit dem Spectrum, ZX81, SAM, Sprinter 2000 oder nahen Verwandten des Sinclair ZX Spectrum befassen, auch Artikel über passende Hardware und Software ist gerne gesehen.
MAC/PC Software: Nur wenn ausdrücklich direkt im Zusammenhang mit den eingangs erwähnten Computern. Sehr gerne: Crosscompiler, Emulatoren, Game Maker und dergleichen. Auf keinen Fall aber Remakes von Spielen alter Plattformen auf moderner Hardware.

Am 12.12. als ich von der JHCon zurückfuhr, hatte ich auf der eisglatten Fahrbahn einen kleinen Ausflug in den Straßengraben. Ein 211 PS starker Hecktriebler verzeiht eben keinen Fahrfehler. Die Onboard-Kamera hat alles schön aufgezeichnet:



Da mich alle zuallererst fragen ob das Auto noch ganz ist: Ja, mir geht es gut. War nicht schnell unterwegs, und das Auto hat keinen einzigen Kratzer. Die Feuerwehr musste es rausziehen. Das hat mich € 150,- gekostet. Also nicht schlimm. Und mir war der Straßengraben lieber als Gegenverkehr.

Ich habe mir vor kurzem den Amstrad CPC 464 gekauft. Ein interessantes Gerät mit Mainboard das sogar in ein Spectrum 48-Gehäuse passen würde. Bin schon gespannt wenn der ZX BASIC Compiler von Boriel auch diese Architektur unterstützt, was ich damit anstellen kann. Die Bildqualität auf meinem Fernseher ist jedenfalls super.
Von Wilko Schröter habe ich ca. 200 Kg Retro-Sachen bekommen. Langsam habe ich ernsthafte Platzprobleme. Aber es waren neben den üblichen Atari/Commodore Sachen auch ein paar interessante ZX81 Sachen dabei. Falls also jemand Busstecker benötigt, bitte mich kontaktieren.

Auf der Wiener Retrobörse habe ich einen TI99/4A Club in Wien kennengelernt. Wenn ein Wiener Leser sich für diese Maschinen interessiert und an einem Clubtreffen teilnehmen will, diese werden hier angekündigt: http://ti99blog.webs.com/

Im Herbst 2011 soll es eine neue Thundercats-TV Serie geben. Der Klassiker wird vermutlich nach einer Versoftung für den Spectrum schreien. Ich meine zwar dass das Original-Spiel schon mal nicht übel war, aber es gibt immer Raum für Verbesserungen, oder andere Genres.



Thunder! Thunder! Thundercats, Hoooo!

Das Chara-Design ist schon mal nicht übel (bis auf Snarf (Dt: Schnuff), der wie ein fetter Pokemon aussieht), ich habe Schlimmeres erwartet, aber es sind noch nicht alle Teammitglieder zu sehen.

Da ich als Jugendlicher ein großer Fan der Serie war, bin ich ziemlich gespannt, ob es meinen Erwartungen gerecht wird. Wie schon bei der ersten Serie, ist es wieder eine Kooperation zwischen japanischen und Studios in den Staaten.
Anders als die früheren Versuche (Kinofilme Live Action oder CG) scheint dieses Projekt zu fruchten, denn es wird von Bandai bereits neues Thundercats-Spielzeug im neuen Design auf den Markt geworfen.
Für alle, die das Design aus den 80ern nicht kennen, das waren die bunten Original-Thundercats.



Nett, oder?

Bleibt nur noch ein Schönes Neues Jahr zu wünschen, auch wenn ziemlich verspätet gegenüber meinem ursprünglich angepeilten Datum.
Hoffentlich kann ich die nächste Ausgabe schneller fertigstellen, aber am 4.4. werden meine Schrauben entfernt.

LCD-Leszek Chmielewski

English summary

We are depended on YOUR articles. I cannot fill any (planed) 24-32 sides all alone, even if I would do it so gladly. Following rules are applicable to sent in articles:
The articles must be for Sinclair computer family itself with the Spectrum, ZX81, SAM Coupé, Sprinter 2000 or near relatives of the Sinclair ZX Spectrum, also article about matching hardware and software likes to be seen.
MAC / PC software: Only if expressly directly in the context of the at the beginning mentioned computers. Very much gladly: Crosscompiler, emulators, Game Maker and such. By no means however remakes of games of old platforms on modern hardware.

At the 12.12 as I returned from JHCon, I had a small trip into the ditch on the icely roadway. A 211 HP Rear-wheel-driven car simply doesn't forgive any driving-mistake. The Onboard-Camera recorded everything beautifully.
Since all first of all ask me whether the car still is whole: Yes, I am fine. I was not very fast, and the car doesn't have any single scratch. The fire brigade had to out-pull it. This costs me € 150. -. Therefore not bad. And the ditch was for me better than meeting the counter-traffic.

I bought myself recently the Amstrad CPC 464. An interesting machine with Main-board which would even fit in a Spectrum 48-casing. Already is tense when the Boriel ZX BASIC compiler will also support this architecture what I can do with it. The picture-quality on my TV is super anyway.
From Wilko Schröter, I have got around 200 kg of Retro-stuff. Slowly, I have serious place-problems. But it besides the usual Atari / Commodore stuff was also a few interesting ZX81 hardware on that occasion. If somebody therefore requires bus-plugs, asks me.

On the Vienna Retrobörse, I met there people who founded a TI99/4A club here. If a Vienna reader wants to be interested in these machines and take part in club-meetings, these are announced here:
http://ti99blog.webs.com/

And now I wish you a happy new year, even if a little bit too late.

LCD

Neues in neuem Jahr

Sinclair kann es nicht lassen

http://www.sinclair-research.co.uk

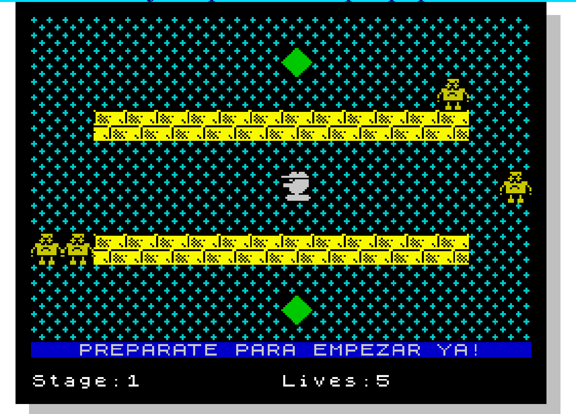
Sir Clive Sinclair hat seine Träume noch nicht zu Grabe getragen. Der C5 ist zwar tot, aber was kommt da auf uns 2011 zu?



Der X-1 ist... erwartungsgemäß ein Elektrofahrzeug. Mit Li-Ion Batterie und Pedalantrieb erreicht das Fahrzeug vermutlich 24 km/h. Der Fahrer sitzt wettergeschützt in der Kabine aus modernsten Materialien. Kosten soll es an die 600 Pfund Sterling. Für Sinclair Fans sicher nicht zu teuer.

Robomas

http://www.ojodepez-fanzine.net/mojoniaplus/viewtopic.php?f=13&t=582



Mit Robore haben die Mojons ein altes Projekt aus der Versenkung geholt. Phantomas und Robotron Crossover. Wie bei Robotron folgen die Roboter dem Phantomas. Man muss sie so geschickt austricksen, dass die dummen Viecher auf den „Diamanten“ stoßen und damit vernichtet werden. Das Spiel ist übrigens in compiliertem BASIC geschrieben, und das Listing wurde im Forum zur Verfügung gestellt. Übrigens sollte man es im 48K oder USR0 Modus starten, weil auch die zwei von 128K BASIC als Befehle reservierten UDGs benutzt wurden. Mit dem Listing kann man es aber leicht ändern, so dass stattdessen Fontzeichen verwendet werden.

Boriels Compiler-the story continues

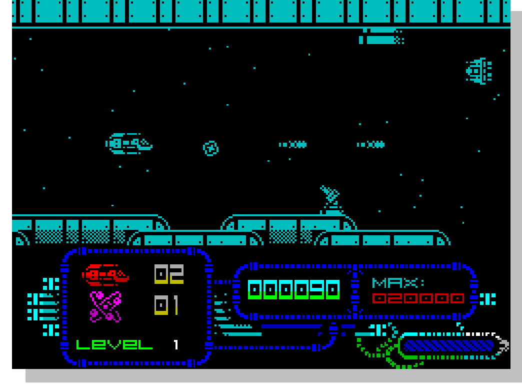
http://www.boriel.com/forum/viewforum.php?f=13

Boriel hat nach längerer Pause (umzugsbedingt) und dem Release der Compilerversion 1.2.6 Stable, angekündigt dass die Arbeiten an Version 2.0.0 begonnen haben. Da Milestone-Releases immer sehr gute Gründe für Versionssprünge haben, ist anzunehmen dass sich sehr viel ändern wird. Es werden vermutlich Multiple Architekturen unterstützt werden (MSX, CPC,...).

Die aktuellste Version ist übrigens 1.2.7 r2087.

Retroworks Genesis

http://www.retroworks.es/php/game_en.php?id=8



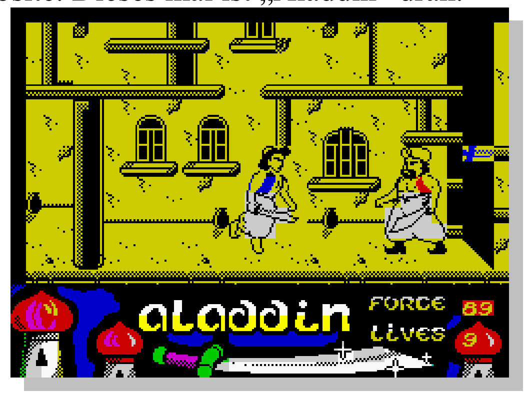
Genesis – Dawn of a new day ist ein neues horizontal scrollendes Shoot'em up (neuenglisch: „Shump“) für den Spectrum 128. Und was für eines! Das Genre galt schon als ausgestorben am Spectrum. Man kann es zwar

nicht mit R-Type vergleichen, doch auch hier ist ein Megablast vorhanden, doch er tötet nur einmal alle Gegner am Bildschirm. Andere Extrawaffen sind: Tripple shoot, Laser, Homing Missile, Bombe (zusätzlich zum normalen Schuss) und Supershoot der in alle Richtungen abgefeuert wird.

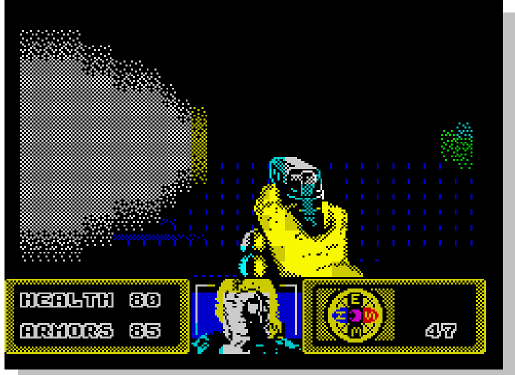
Oleg Origin

http://zx.pk.ru/showthread.php?t=12804&page=4

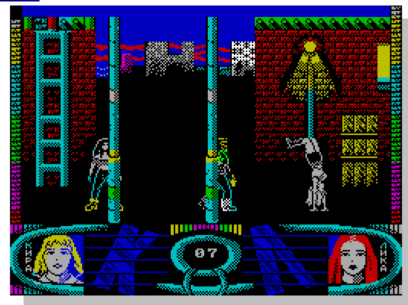
Oleg hat ein paar weitere seiner alten Projekte präsentiert, auch wenn noch nicht auf seiner Website. Dieses mal ist „Aladdin“ dran.



Sowie „Quake (The Dark)“. Beide sind als TRD Versionen verfügbar.



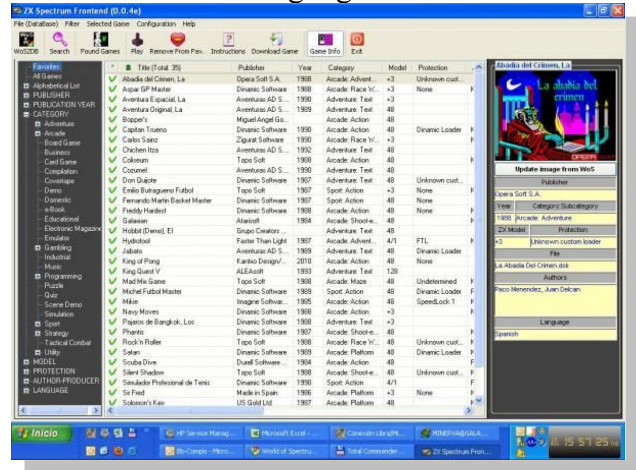
Ein weiterer bislang unbekannter und nicht fertig gestellter Titel ist Wild Angels: http://www.worldofspectrum.org/infoseekid.cgi?id=0024889



WOS Plugin

http://programbytes48k.wordpress.com/2010/11/24/wos-zx-spectrum-frontend-v0-0-4e/

Wenn man die Infoseek Informationsbibliothek in seinen eigenen Programmen benutzen möchte, so ist dies neuerdings problemlos möglich, und es wurde bereits ein WOS Infoseek Frontend entwickelt, welches eben dieses neue WOS Feature nutzt um Informationen für Spiele für bestimmte Emulatoren zur Verfügung zu stellen.



Reversao

http://www.worldofspectrum.org/infoseekid.cgi?id=0025316



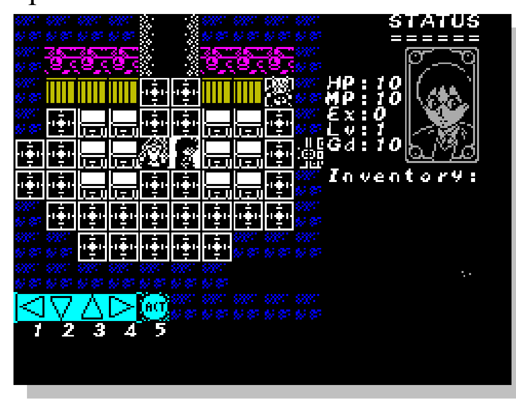
Dieses neue Spiel erinnert etwas an Türme von Hanoi, aber dann ist es doch wieder anders. Man muss Türme-von-Hanoi-mäßig Eine Figur in eine andere umbauen (welche in welche, das kann man sich aussuchen).

Harry, hol schon mal den Zauberstab

http://www.worldofspectrum.org/infoseekid.cgi?id=0025328

C.M.Gilles aus Deutschland hat wieder ein

neues Harry Potter Spiel produziert: Harry the Magical – The Hero Hallows. Wie die bisherigen Spiele ist auch dieses in BASIC geschrieben, aber trotzdem leider nur als Z80 Snapshot eingeschickt worden. Kein TAP, kein TZX... Das ist wirklich schade, denn schlecht ist das Spiel nicht.



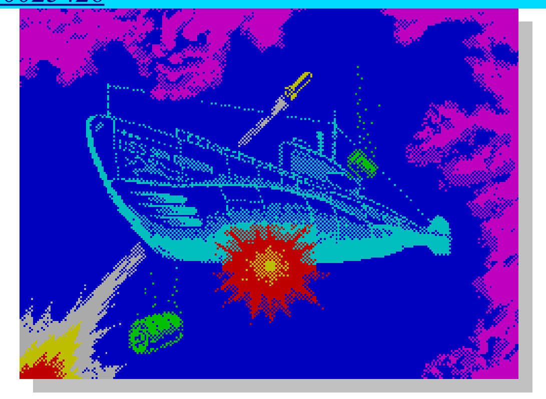
Neues QL Forum

http://qlforum.co.uk/

Es wurde ein neues englischsprachiges QL Forum eröffnet, mit sehr interessanten Beiträgen, z.B. wie man Zork auf dem QL zum laufen bringt.

Seedrache ist fertig

http://www.worldofspectrum.org/infoseekid.cgi?id=0025426



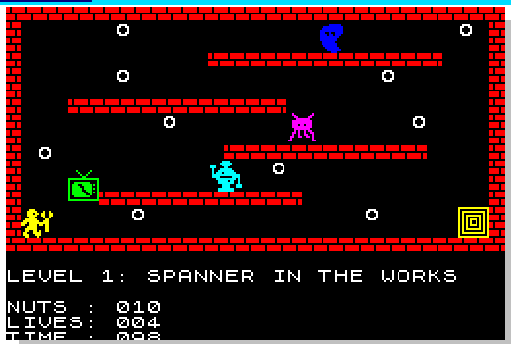
Letztes mal habe ich hier über die Spectrum Konvertierung von Atari berichtet. Nun ist Seadragon fertig und verwöhnt die Augen durch butterweiches Softscrolling und Unterstützung diverser Soundsysteme wie AY oder General Sound. Die Ohren werden mit „Das Boot" verwöhnt. Das Gehirn wird dadurch verwöhnt, dass der Sourcecode veröffentlicht wurde, so kann jeder sehen wie das Softscrolling arbeitet.

Sid Spanners

http://www.worldofspectrum.org/infoseekid.cgi?id=0025409

http://www.worldofspectrum.org/infoseekid.cgi?id=0025461

http://www.worldofspectrum.org/forums/showthread.php?t=32308



Digital Prawn hat nach „Smiler", welcher mit „Smiler's Christmas Sack" wohl gerade seinen letzten Teil bekommen hat, eine neue Serie angefangen. „Sid Spanners" erinnert stark an „Manic Miner", und es sind bereits zwei Teile erschienen, die mit beinahe derselben Engine funktionieren.

Crimbo

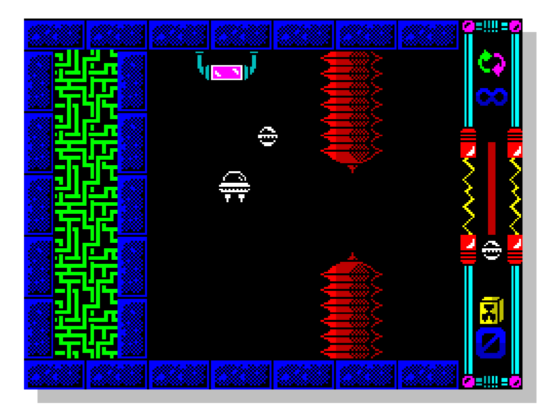
http://www.worldofspectrum.org/infoseekid.cgi?id=0025374



„Crimbo – A Gloop Troops Tale" ist wie der Untertitel es bereits andeutet, der Nachfolger von „Gloop Troops" (Test in letzter Ausgabe), und mit sehr weihnachtlicher Stimmung unterlegt. Technisch 1A! Absolut empfehlenswert.

Phaeton

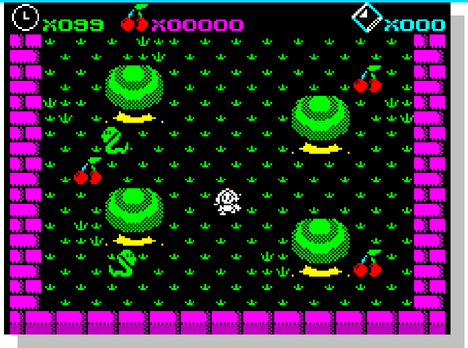
http://www.worldofspectrum.org/infoseekid.cgi?id=0025399



Ralf aus Polen hat mit Phaeton ein revolutionäres (im wahrsten Sinne des Wortes) Spiel geschrieben. Man steuert ein UFO (eine Anspielung auf „Saucer“?) mit zwei Satelliten, und darf mit ihnen nicht allzu oft anecken. Schwer zu beschreiben aber ein geniales Spiel das man einfach selbst ausprobieren muss..

Forest Raider Cherry

http://www.worldofspectrum.org/infoseekid.cgi?id=0025408



Ist zwar nicht Lara Tomb Raider, macht aber auch Spaß. Timmy verpackte wieder familienfreundliche Grafik statt Mojons nackter Tatsachen in ein Cheril- ähnliches Spiel. Macht sehr viel Spaß.

Herbert the silent

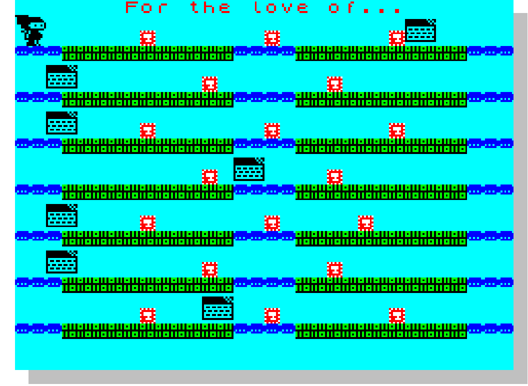
http://www.worldofspectrum.org/forums/showthread.php?t=32538



Bob hat das derzeit noch soundlose „Herbert the Turbot“ veröffentlicht. Er wartet noch bis ein passender Soundtrack reinkommt. Das Spiel ist aber an sich bereits fertig. Ja, ich bin nicht so dass ich sage dass ein Spiel ohne Sound keine Daseinsberechtigung hätte.

Briefmarkensammler

http://www.worldofspectrum.org/infoseekid.cgi?id=0025468

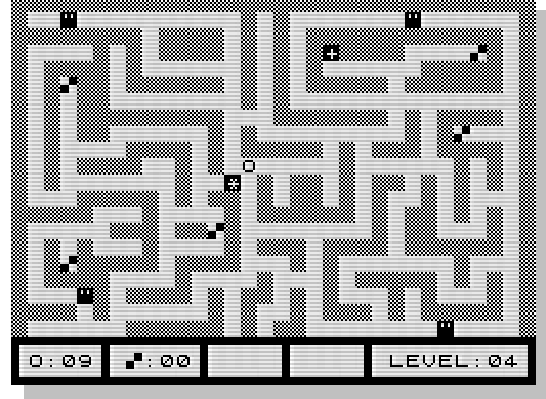


Dave Hughes hat ein neues Spiel (sein erstes in M/C) geschrieben. Man spielt einen intergalaktischen Briefmarkensammler. Das Spiel ist schwer und sehr schnell.

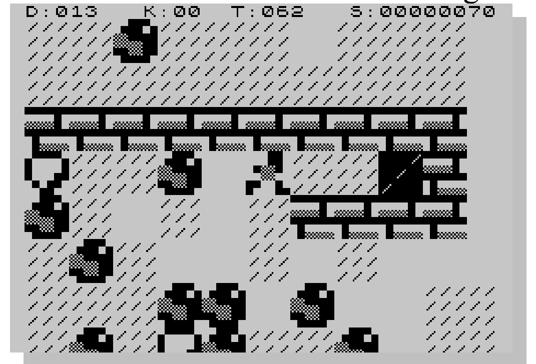
Zeddy News

http://forum.tlienhard.com/phpBB3/viewtopic.php?f=2&t=459
http://forum.tlienhard.com/phpBB3/viewtopic.php?f=6&t=482

Ja, auch der ZX81 kommt dieses mal zum Zug. Bbock hat das Spiel „Laby“ für 16K (war für 64K geplant, aber z88dk scheint etwas dagegen gehabt zu haben, so dass alles abstürzte) geschrieben und arbeitet derzeit fleißig an der Spectrum Version. Die Z88kd-Sources sind verfügbar.



Auch Bob Smith hat neuen ZX81 Stuff gecodet:



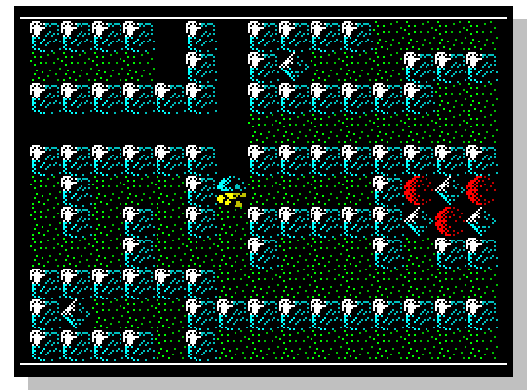
http://www.bobs-stuff.co.uk/boulderlogic.html

„Boulder Logic ist ein „Boulder Dash“ Klon für 16K, mit Scrolling. Das Character-basierende Bild des ZX81 erlaubt überraschend schnelle scrollende Spiele.

Minenarbeiter

http://www.worldofspectrum.org/infoseekid.cgi?id=0025367

Wo wir gerade noch bei „Boulder Dash“ waren...



„Mine Worker“ ist ein sehr lieb programmierter schneller Klon aus Rußland. Leider hat der Programmierer, wie andere User festgestellt haben, nicht daran gedacht den Punktestand im Spiel anzuzeigen. Meine Meinung: Punkte sind doch sowieso unwichtig, Hauptsache ist dass man ins nächste Level kommt. Von daher ist es mir egal.

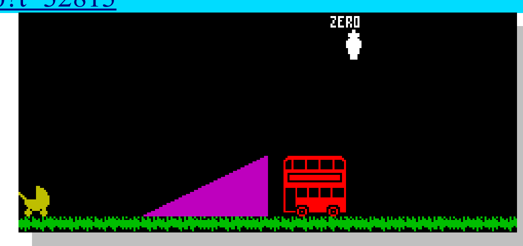
Exominierrren!!!

http://hem.bredband.net/magli143/exo/

Nach beinahe 3 Jahren seit dem Erscheinen des Exonomizer Packers v. 2.0 Beta 7, ist endlich die Final + 2.0.1 releast worden. Dieser PC-basierende Extrem-Packer ist zwar für C64/Atari XL/XE/Apple][gedacht, aber dank Metalbrain gibt es auch einen Z80 Decruncher dafür (nur RAW und backward modes), was bedeutet dass auch Spectrum, MSX, Enterprise und Amstrad CPC-Benutzer einige der Vorteile des Packers nutzen können. Der Z80 Decruncher ist als ASM Source in der ZIP Datei inkludiert.
Der MegaLZ Packer ist zwar eine Alternative, aber im Moment scheint die Webseite mit einem Trojaner infiziert zu sein, und mein Virenscanner lässt mich da nicht rein. Mit einer Virtualisierung oder Sandbox würde ich mich aber durchaus trauen die Datei herunterzuladen (die Datei selbst ist nicht infiziert).

Tippfehler

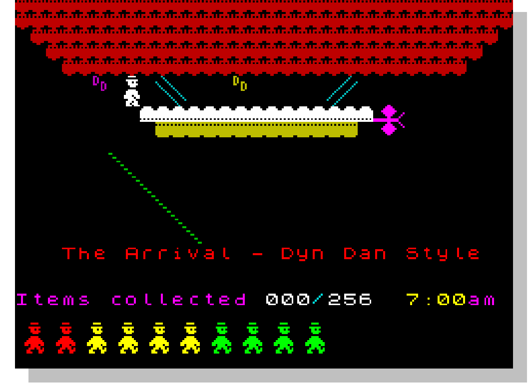
http://www.worldofspectrum.org/forums/showthread.php?t=32813



Mit Utter Tripe gibt es wieder ein neues Spiel von Jonathan Cauldwell, in dem der Coder die Tradition der vielen (40) ungewöhnlichen Spielkonzepte in einem Spiel fortsetzt.
Hier muss man in verschiedenen Minispielen beweisen wie schnell man am Spectrum tippen kann. Für Cronosoft wird ein Remix gemacht.

JSW2010

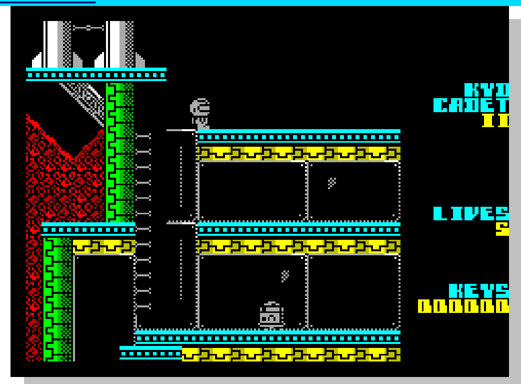
http://www.worldofspectrum.org/infoseekid.cgi?id=0025421



Jet Set Willy, the 2010 Megamix ist der Nachfolger der 2005 Version. Nun verfügbar.

Kyds Rettungsmission

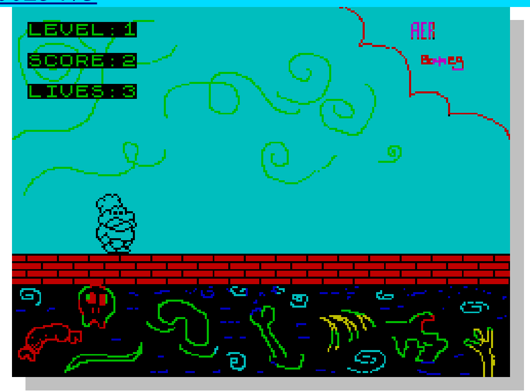
http://www.worldofspectrum.org/infoseekid.cgi?id=0025269



„Kyd Cadet II: The Rescue of Pobbleflu“ ist der Nachfolger zu Kyd Cadet von Paul Jenkinson, welcher vor ein paar Monaten erschienen ist. Sehr schöner Plattformer.

Panic now!

http://www.worldofspectrum.org/infoseekid.cgi?id=0025473



„Panic Restaurant“ ist ein neues Game von AER, und leider in der bewährten AER-Qualität. Wobei ich zugeben muss dass von mal zu mal die Spiele besser werden. In diesem Spiel rennt man nach vorne los und überspringt Löcher im Boden.

Fischer Fritz Frisst frische Fische

http://zx.pk.ru/showthread.php?t=14837

Alko (was für ein Pseudonym für einen russischen Coder!!!) schreibt an einem Spiel bei dem es um Fische und Fressen geht. Nichts für Vegetarier, jedoch spielt man selbst einen Fisch. Es sind bereits einige Entwicklungssnapshots erschienen und das Spiel wird in BASIC programmiert, dennoch sehr interessant.



Joytape

http://www.freeweb.hu/csabagondos/joytape/joytape.html



Joytape ist ein Flash-basierender TZX-Abspieler der wie ein Joystick aussieht, und am Tapeeingang des Spectrum angeschlossen wird. Es sind 6 Spiele von WSS inkludiert, sowie ein offenbar bisher unveröffentlichtes Bonus-Spiel. Diese werden mit normaler Geschwindigkeit geladen.

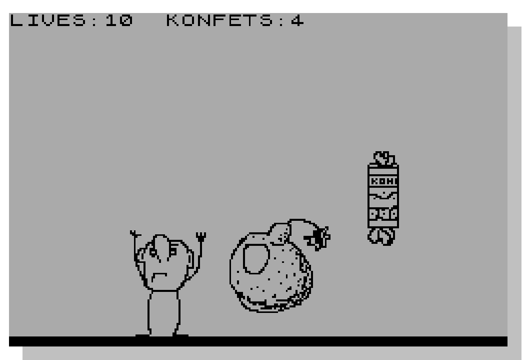


Die Batterien sollten für etwa 1000-malige Benutzung reichen.

Wenn Süßes vom Himmel fällt...

http://www.worldofspectrum.org/infoseekid.cgi?id=0025488

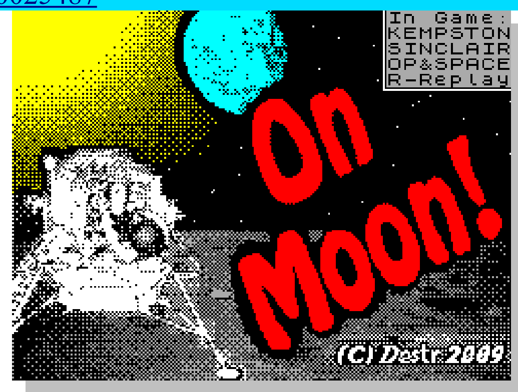
Nein, es geht hier nicht um Zuckerl-Onkel die Süßigkeiten selbstlos an kleine Mädchen verteilen, sondern um ein neues Spiel in AER-Qualität.



Im Spiel „Konfets“ fallen vom Himmel Süßigkeiten und Bomben. Die einen sollte man aufsammeln, die anderen vermeiden.

Der Eagle ist gecrasht

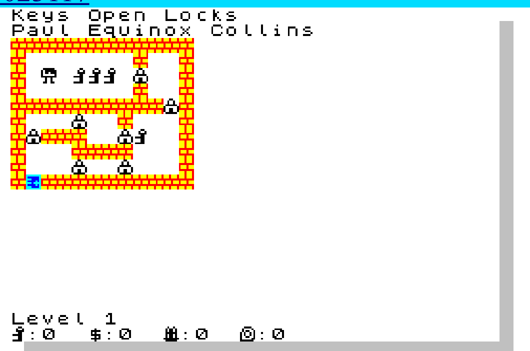
http://www.worldofspectrum.org/infoseekid.cgi?id=0025487



Descructor und Newart haben ein neues Spiel releast: „On Moon!“. Die Landung auf dem Mond ist nicht leicht, also was liegt näher als ein Spiel darüber zu schreiben. Ist jedenfalls gelungen.

Pilze! Sie sind alle gegen mich!

http://www.worldofspectrum.org/infoseekid.cgi?id=0023117



Dr. BEEP (nicht verwechseln mit Mr. Beep oder Doctor Who) hat die 2011-Version von Mushroom Man für Spectrum verbessert. Die alte Version war für 128+/+2 geschrieben. Sie funktionierte nicht auf dem Spectrum +2A/+3 solange man nicht das 128K+ ROM eingebaut hat. Die neue Version sollte auch am 48K funktionieren.

Spektakulär?

http://www.spectaculator.com/news/010211.php

Jonathan Needle hat die Version 7.5 des kostenpflichtigen ZX Spectrum Emulators „Spectaculator“ freigegeben.
Natürlich gibt es wieder zahlreiche Verbesserungen für die zahlenden Kunden.

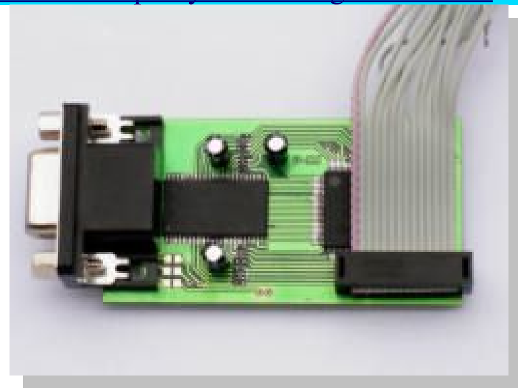
Programmerweiterung

http://cronosoft.orgfree.com/

Cronosoft erweitert sein Portfolio um weitere Computersysteme für die neue Programme vertrieben werden sollen: „Sord M5“, „Aquarius“,“TI99/4A“ und „Laser 200“ wurden als neue Plattformen ins Programm aufgenommen.

ZX Scandoubler

http://velesoft.speccy.cz/zx/zx-vga/index.html

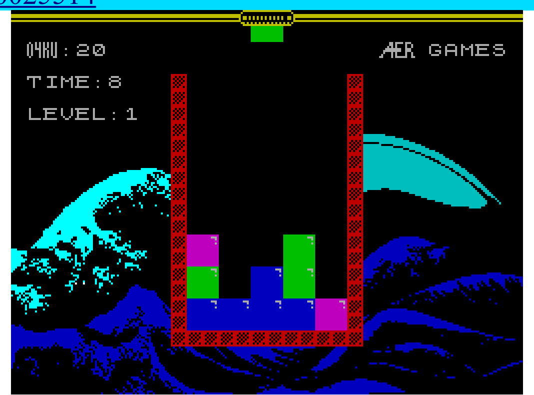


Von Velesoft kommt ein neuer interner Scandoubler, mit dem es möglich ist, einen VGA Monitor an den Spectrum ohne Converter anzuschließen, was in hervorragender Qualität resultiert. Eine Version für SAM Coupé wäre auch möglich.
Verkauft wird das Interface vermutlich nicht, denn es sind nur Platinenlayout und CPLD-Programme zum Herunterladen verfügbar. Es erfordert SMD-Lötausrüstung um das ganze aufzubauen, und die Leitungen müssen sicher an die ULA angelötet werden. Zudem erfordert es

für jede Spectrum-Version eigene CPLD-Programme und Einbauanleitungen.

Russische Kristale

http://www.worldofspectrum.org/infoseekid.cgi?id=0025514



AER produziert inzwischen wohl mehr Spiele als Mojon Twins, was aber leider zu Lasten der Qualität geht. Mit „Crystal Cubes“ ist eine Coloris-Variante entstanden, die an sich nicht übel ist, auch wenn die 3er Kombinationen nur Vertikal erkannt werden, nur die Steuerung ist etwas schwammig und vor allem, in Emulatoren auf Rechnern mit deutscher Tastatur, etwas problematisch (Z und Y sind ja vertauscht).

RetroWiki Magazin

http://www.retrowiki.es/news.php?extend.48.6



Die Zero-Ausgabe des spanischen Retro-Magazins „RetroWiki“ ist erschienen kann nun kostenlos heruntergeladen werden.

ZX Spectrum Relaunch

http://www.telegraph.co.uk/technology/video-games/8304237/ZX-Spectrum-relaunch-gaming-goes-back-to-the-future.html



Ist ja fast schon April, Zeit für einen Spectrum Relaunch, dachte sich wohl Elite, und kündigte das an: Ein Bluetooth-Keyboard in Form des ZX Spectrum, als Tastatur für Smartphones auf denen Spectrum Emulatoren laufen.
Also bitte, echte Hardware auf einem modernen in modernen Herstellungstechniken gefertigten Spectrum-Mainboard, dazu diverse eingebaute Erweiterungen wie Joystickinterface, USB-Ports, SD-Slot. Das würde ich verstehen und das Gerät kaufen (natürlich nur wenn der Spectrum nicht emuliert ist), aber Tastaturgadget? Wer zu viel Geld hat, bitte...

Wir sind Kaiser!

http://www.sinclair-spectrum.de

FUERST 5012
LCD von WUERTTEMBERG
1.Jahr Ihrer Herrschaft
ANNO DOMINI 1700
100% Vorrat Jahr 1699 verbraucht
Normale Ernte
Ernteertrag: 18000 kg
Kornvorrat: 18000 kg
Kornbedarf: 11351 kg
Kornpreis/100kg 67
Landpreis/100ha 219
Landbesitz: 10000 ha
Vermoegen: 2500
Sie koennen: 1=Korn kaufen
2=Korn verkaufen
3=Land kaufen
4=Land verkaufen
5=Weiter

Ebi Häger hat endlich seine hervorragende Umsetzung von „Kaiser“ endlich zum Download als TZX freigegeben. Super-Spiel! Ich liebe es.

Die nützlichsten ROM - Routinen

Die nützlichsten ROUTINEN 1987 written by Klaus Jahn.

Sie haben sich bestimmt schon darüber geärgert, dass Sie In Maschinensprache auf den BASIC-Komfort verzichten mussten?
Das soll sich nun auch für Sie ändern. Ich gebe Ihnen hier eine komplette Auflistung der gesamten nützlichen ROM-Routinen im Spectrum-ROM.
Außerdem nenne ich Ihnen die Anwendung mit den Registern, die die Routinen benutzen. Sie werden sehen, dass einige Routinen mit dem so genannten Calculatorstack arbeiten; dieser ist eine Art Stapel, um 5-Byte-Fließkommazahlen oder Stringparameter abzuspeichern, aber dazu später mehr!
Ich zeige Ihnen jetzt die wichtigsten Routinen:

BEEP x/y,z:
a) ld a,{x} Dividend der Tonlänge
call 11560 auf Calculatorstack legen
b) ld a,{y} Divisor der Tonlänge
call 11560 auf Calculatorstack legen
c) rst 40 Calculatoraufruf
Byt. #5 Zahlen dividieren
Byt. #56 Ende des Calculators
d) ld a,{z} Tonhöhe
call 11560 auf Calculatorstack legen
e) call 1016 BEEP-Routine aufrufen.

BORDER x:
a) ld a,{x} Farbe in Akku bringen
b) call 8859 BORDER-Routine aufrufen.

BRIGHT x: [siehe "Farben setzen"]

CIRCLE x,y,z:
a) ld a,{x} X-Koordinate
call 11560 auf Calculatorstack legen
b) ld a,{y} Y-Koordinate
call 11560 auf Calculatorstack legen
c) ld a,{z} Radius
call 11560 auf Calculatorstack legen
d) exx
push hl hl' - Register sichern
exx
e) call 9005 CIRCLE-Routine aufrufen
f) exx
pop hl hl' - Register holen.
exx

CLS: call 3435

COPY: call 3756

DRAW x,y:
a) ld a,{x} X-Koordinate
call 11560 auf Calculatorstack legen
b) ld a,{y} Y-Koordinate
call 11560 auf Calculatorstack legen
c) exx
push hl hl' - Register sichern
exx
d) call 9335 DRAW I - Routine
e) exx
pop hl hl' - Register holen.
exx

DRAW x,y,z:
a) X- und Y- Koordinaten wie bei DRAW x,y auf den Calculatorstack bringen
b) ld a,{z} Radiantenzahl call 11560 auf Calculatorstack legen
c) hl' -Register sichern (siehe DRAW x,y)
d) call 9108 DRAW II - Routine
e) hl' -Register holen (Siehe DRAW x,y).

FLASH: [siehe "Farben festlegen"]

INK: [siehe "Farben festlegen"]

INVERSE: [siehe "Farben festlegen"]

LIST x:
a) ld hl,{Startadresse von Zeile x}
b) call 6229

LOAD Header:
a) ld ix,{Anfangsadresse des Headers}
b) ld de,17
c) xor a
d) scf

e) call 1366   LOAD-Routine

LOAD CODE:
a) ld ix,{Anfangsadresse des Codes}
b) ld de,{Länge des Codes}
c) ld a,255
d) scf
e) call 1366   LOAD-Routine

NEW: call 4535

OVER: [siehe "Farben festlegen"]

PAPER: [siehe "Farben festlegen"]

PAUSE x:
a) res 5,(iy+1) Eingabe freimachen
b) ld bc,{x}   Länge in msec. nach bc
c) call 7997

PLOT x,y:
a) ld c,{x}   X-Koordinate in c
b) ld b,{y}   Y-Koordinate in b
c) call 8927   PLOT-Routine

PRINT #x:
Bevor man irgendetwas zum ersten mal ausdruckt, muss der Bildkanal mit PRINT# geöffnet werden ! Das geschieht so:
a) ld a,{x}   Kanalnummer in Akku
b) call 5633   Kanal öffnen
Die Kanäle lauten:
#3 Ausdruck auf Drucker
#2 Ausdruck auf Screen Zeile 0 bis 21
#1 Ausdruck auf Screen Zeile 22 bis 23
#0   -//-
#255 Workspace
#254 Keyboard

PRINT CHR$(x):
a) ld a,{x}   Code des Zeichens in Akku
b) rst 16   Ausdruck

PRINT a$:
a) ld de,{Startadresse der Zeichenkette}
b) ld bc,{Länge der Zeichenkette}
c) call 8252   Textdruck.

PRINT x: {x liegt im Bereich 0 - 9999}
a) ld bc,{x}   Zahl in bc-Register bringenb)
call 6683   Zeilennummerdruck-Routine

PRINT x:
a) ld bc,{x}   Zahl
b) call 11563 auf Calculatorstack legen
c) call 11747 Zahlenausdruck.

PRINT AT y,x:
a) ld b,{24-y}
b) ld c,{33-x}
c) call 3545

SAVE Header:
a) ld ix,{Startadresse des Headers}
b) ld de,17
c) xor a
d) call 1218   SAVE-Routine

SAVE CODE:
a) ld ix,{Startadresse des Codes}
b) ld de,{Länge des Codes}
c) ld a,255
d) call 1218   SAVE-Routine.

SCROLL: call 3282

STOP: call 7406

VERIFY CODE:
a) ld ix,{Startadresse des Codes}
b) ld de,{Länge der Codes}
c) ld a,255
d) and a
call 1366   LOAD-Routine

SOUND x,y:
(Erzeugt einen kurzen Pfeifton. x ist die Tontiefe und liegt im Bereich
[1..65535]. y ist die Tonlänge und
liegt im Bereich [0..65535].)
a) ld hl,{x}
b) ld de,{y}
c) call 949

BASIC-Programm löschen: call 4656

Calculatorstack löschen: call 5829

Ausdruck des obersten Calculator-stackwertes: call 11747

Akkuinhalt auf Calculatorstack legen: call 11560

BC - Registerinhalt auf Calculatorstack legen: call 11563

Obersten Calculatorstackwert in den Akku bringen: call 11733

Obersten Calculatorstackwert ins bc-Registerpaar bringen: call 11682

FARBEN FESTLEGEN:
Als Beispiel geben wir uns an, wie die Befehle PAPER 7: INK 9: OVER 1 in Assembler aussehen:
Die normale temporäre Farbzuweisung geschieht ganz einfach mit rst 16.
Soll z.B. PAPER 7 ausgedrückt werden,
so schreibt man:
a) ld a,17 (=ASCII-Code von PAPER)
b) rst 16
c) ld a,7 (für PAPER 7)
d) rst 16
Sollen die Farben allerdings bleiben
(nach CLS oder PRINT #), dann muss man mit folgenden call-Befehlen arbeiten:
a) call 3405
b) Festlegung der gewünschten Farbanweisung mit rst 16
c) call 7341

MEM-Anweisung:
a) call 7962
b) ld hl,65535
c) and a
d) sbc hl,bc
e) der in BASIC noch verfügbare Bereich steht jetzt im hl-Register.

EDIT: call 3884

Ausgabe eines Listings: call 6037

hl = hl * de: call 12457

Bei folgenden 3 Prüfroutinen trifft die Bedingung zu, wenn das Carry-Flag=0 ist:
Akku auf Zifferncode prüfen: call 11547
Akku auf Buchstabencode prüfen: call 11405
Akku auf alphanumerisch prüfen: call 11400

Ausgabe von Text, dessen letztes Byte mit gesetztem bit 7 markiert ist:
a) ld de,{Startadresse des Textes}
b) call 3082

BASIC-Zeilenkursor hoch: call 4185

BASIC-Zeilenkursor runter: call 4083

Löschen von Calculatorstack, Workspace und Editorspace: call 5808

Berechnen einer Bildschirm-Pixel-Adresse nach HL. Angegeben wird die (X,Y)-Pixelposition:
a) ld c,{X-Koordinate}
b) ld b,{Y-Koordinate}
c) call 8874
d) hl gibt die SCREEN$-Adresse an.

a = c + 10 * a: call 12171

Scrollen einer bestimmten Bildzeilen-zahl (von unten gesehen):
a) ld b,{Zeilenzahl}
b) call 3584

Löschen einer bestimmten Bildzeilen- zahl (von unten gesehen):
a) ld b,{Zeilenzahl}
b) call 3652

Ausgabe einer Fehlermeldung:
Dazu wird der Befehl rst 8 benutzt. Der Code der Fehlermeldung wird einfach als ein Byte dem rst 8-Befehl angefügt.
Man befindet sich nach aufruf dieses Befehls sofort wieder im BASIC. Z.B. wird die Meldung 0 OK so gemacht:

a) rst 8
b) Byt. #255 {Meldecode minus 1}

POINT (x,y):
a) ld c,{x}
b) ld b,{y}
c) call 8910
d) call 7833
e) Ergebnis (1 oder 0) befindet sich im Akku.

SCREEN$ (y,x):
a) ld a,{y} Y-Koordinate
call 11560 auf Calculatorstack legen
b) ld a,{x} X-Koordinate
call 11560 auf Calculatorstack legen
c) call 9525 SCREEN$-Routine
d) call 11249 String-Parameter vom Calculator-stack holen
Jetzt befinden sich die Stringparameter (Länge und Startadresse) in den Registern BC und DE. Um das Ergebnis der Funktion SCREEN$ zu bekommen, schreibt man einfach:
e) ld a,(de)
und das Ergebnis (der ASCII-Code) befindet sich im Akku.

ATTR (y,x):
a) ld c,{y}
b) ld b,{x}
c) call 9603
d) call 7833
e) Das Ergebnis steht im C-Register.

Stringdefinition:
Man kann einen String schaffen, indem man seine Startadresse ins de- und seine Länge ins BC-Register bringt.
Der Akku muss 0 sein; dann werden diese 5 Bytes mit der Routine
call 10929
auf den Calculatorstack gebracht.
Mit der Routine
call 11249
hingegen kann man ein Stringparameter vom Calculatorstapel herunter holen.

Der Calculator:
Wie man mit dem rst 40-Befehl umgeht, haben Sie ja schon bei dem BEEP-Befehl gesehen. Der Calculator kann immer nur den obersten oder die obersten 2 Calculatorstack-Elemente behandeln. Das Ergebnis wird dann wieder auf den Stack zurückgelegt. Ich gebe Ihnen jetzt eine komplette Auflistung der so genannten Calculator-opcodes, die die sämtlichen Spectrum-Funktionen darstellen.

Diese Opcodes werden einfach als Bytes hinter dem rst 40-Befehl angefügt.

Im Folgenden sehen Sie eine Liste, in der die Calculatoropcodes, also sämtliche BASIC-Funktionen des Spectrum aufgeführt sind. Bei jedem Opcode sehen Sie auch die Calculatorsack-Manipulation durch die Operationen. In der nun folgenden Liste ist immer das rechteste Calculatorstack-Element das oberste auf dem Stack liegende Element !

Opcode	Operation	Stack vorher	/Nachher
1)	tauschen:	X,Y,Z	X , Z , Y
2)	löschen	X,Y,Z	X,Y
3)	-	X,Y,Z	X ,(Y - Z)
4)	*	X,Y,Z	X ,(Y * Z)
5)	/	X,Y,Z	X ,(Y / Z)
6)	^	X,Y,Z	X ,(Y ^ Z)
15)	+	X,Y,Z	X ,(Y + Z)
31)	SIN	X,Y,Z	X,Y,(SIN Z)
32)	COS	X,Y,Z	X,Y,(COS Z)
33)	TAN	X,Y,Z	X,Y,(TAN Z)
34)	ASN	X,Y,Z	X,Y,(ASN Z)
35)	ACS	X,Y,Z	X,Y,(ACS Z)
36)	ATN	X,Y,Z	X,Y,(ATN Z)
37)	LN	X,Y,Z	X,Y,(LN Z)
38)	EXP	X,Y,Z	X,Y,(EXP Z)
39)	INT	X,Y,Z	X,Y,(INT Z)
40)	SQR	X,Y,Z	X,Y,(SQR Z)
41)	SGN	X,Y,Z	X,Y,(SGN Z)
42)	ABS	X,Y,Z	X,Y,(ABS Z)
43)	PEEK	X,Y,Z	X,Y,(PEEKZ)
44)	IN	X,Y,Z	X,Y,(IN Z)
49)	dupliz.	X,Y,Z	X, Y, Z, Z
50)	MOD/DIV	X,Y	Rest , (X/Y)
56)	End Calc.		
160)	0	X,Y	X,Y, 0
161)	1	X,Y	X,Y, 1
162)	0.5	X,Y	X,Y, 0.5

163)	PI/2	X,Y	X,Y, 1.5707
164)	10	X,Y	X,Y, 10
8)	AND	X,Y,Z	X,(Y AND Z)
7)	OR	X,Y,Z	X,(Y OR Z)
48)	NOT	X,Y	X , (NOT Y)
58)	Int.-Teil schneiden	X,Y,Z	X,Y, Z-INT Z

Es gibt aber hier auch die Funktionen, die mit Strings arbeiten; z.B. wandelt die STR$-Funktion einen numerischen Ausdruck in einen Stringausdruck um. Wenn man das mit dem Calculator tut, so muß man ein Stringparameter auch als solches vom Calculatorstack herunterholen und nicht etwa als Zahl.
Sie erinnern sich: Mit call 11249 kann man ein Stringparameter, welches auf dem Calculatorstack liegt, in die Register bc (die Stringlänge) und de (die Start-Adresse des Strings) bringen. Umgekehrt kann man die Inhalte dieser beiden Registerpaare als Stringparameter mit call 10929 auf den Calculatorstapel~bringen.
Sehen Sie nun auch diese Funktionen:

Opcode	Operation	Stack vorher	/Nachher
30)	LEN	X , Y$	X , LEN Y$
46)	STR$	X,Y,Z	X,Y,STR$ Z
47)	CHR$	X,Y,Z	X,Y,CHR$ Z
25)	USR	X , Y$	X , USR Y$
28)	CODE	X , Y$	X , CODE Y$
23)	String +	X , Y$, Z$	X , Y$ + Z$
27)	negieren	X,Y,Z	X , Y , (-Z)

Des Weiteren gibt es Testfunktionen, die als Ergebnis entweder eine 1 (wahr) oder eine 0 (falsch) liefern. Um Speicherplatz zu sparen, wurden alle 12 Vergleichsoperationen in eine Routine gepackt. Zur Unterscheidung dient in dem Falle das b-Register, denn man muss hier den Opcode zusätzlich ins b-Register laden.
In der folgenden Tafel sind diese Operationen mit , b gekennzeichnet. Beachten Sie, daß es 6 Vergleichsoperationen für numerische Operanden und 6 für Stringoperanden gibt !

Opcode	Operation	Stack vorher	/Nachher
9 ,b	<=	X , Y	1 wenn X<=Y, 0 wenn X>Y
10 ,b	>=	X , Y	1 wenn X>=Y, 0 wenn X<Y
11 ,b	<>	X , Y	1 wenn X<>Y, 0 wenn X=Y
12 ,b	>	X , Y	1 wenn X>Y, 0 wenn X<=Y
13 ,b	<	X , Y	1 wenn X<Y, 0 wenn X>=Y
14 ,b	=	X , Y	1 wenn X=Y, 0 wenn X<>Y
17 ,b	<= String	X$, Y$	1 wenn X<=Y$, 0 wenn X$>Y$
18 ,b	>= String	X$, Y$	1 wenn X>=Y$, 0 wenn X$<Y$
19 ,b	<> String	X$, Y$	1 wenn X<>Y$, 0 wenn X$=Y$
20 ,b	> String	X$, Y$	1 wenn X$>Y$, 0 wenn X<=Y$
21 ,b	< String	X$, Y$	1 wenn X$<Y$, 0 wenn X>=Y$
22 ,b	= String	X$, Y$	1 wenn X$=Y$, 0 wenn X<>Y$
54	< 0	X	1 wenn X<0, 0 wenn X>=0
55	> 0	X	1 wenn X>0, 0 wenn X<=0

Ausrechnen eines Terms, der als String im Speicher steht:
Dazu benutzt man die VAL/VAL$-Routine.
Sie beginnt ab Adresse 13790; die Unterscheidung von VAL und VAL$ erfolgt wieder über das b-Register.
VAL hat den Operationscode 29
VAL$ hat den Operationscode 24.
Man kann z.B. mit der VAL-Routine eine mathematische Aufgabe lösen, ähnlich PRINT VAL "12 * SIN (n/PI)". Nehmen wir an, unsere Aufgabe steht ab Adresse 60000 und ist 16 Bytes lang. So schreiben wir:
a) ld de,60000 {Startadresse}
b) ld bc,16 {Länge}
c) xor a
d) call 10934 Parameter auf Calc.stack
e) ld b,29 Opcode von VAL nach b
f) call 13790 VAL/VAL$-Routine
g) Das Ergebnis liegt auf dem Calculatorstack.

Und nun zum Schluss möchte ich Ihnen noch

zeigen, wie man mit dem rst 40-Befehl umgeht. Man kann nämlich mehrere Operationen hintereinander ausführen. Nehmen wir z.B. an, wir wollten den Term
10 + 23 * SIN (x / 80 * PI)
mit dem Calculator berechnen.

Wir müssten zuerst x auf den Calculator-Stack legen und dann von innen nach außen fertig rechnen:

```
a) X liegt auf Stack          X
b) ld a,80
c) call 11560                 X , 80
d) rst 40
   Byt. #5    /               X / 8
   Byt. #163  PI/2            (X/80) , PI/2
   Byt. #49   DUPLICATE       (X/80) ,PI/2,PI/2
   Byt. #15                   (X/80)   ,PI/2+PI/2
              (X/80) , PI
   Byt. #4                    (x/80) * PI
        (x/80*PI)
   Byt. #31   SIN             SIN (x/80*PI)
   Byt. #56   End.
e) ld a,23
f) call 11560                 SIN(x/80*PI)  ,  23
        SIN(x/80*PI) , 23
g) rst 40
   Byt. #4      *             23*SIN(x/80*PI)
   Byt. #164    10          23*SIN(x/80*PI) , 10
   Byt. #15     +           10+23*SIN(x/80*PI)
   Byt. #56   End.
h) ret
```

i) das Ergebnis liegt dann auf dem Calculator-stack !

Neben den BASIC-Funktionen stehen dem Calculator auch noch 6 Speicher zu Verfügung, in die er Werte bringen kann. Diese 6 Speicher werden wie folgt bezeichnet:"
MEM0 , MEM1 , MEM2 , MEM3 , MEM4 , MEM5

Um eine Kopie des obersten Calculator-stackwertes in eine dieser Speicherzellen zu bringen, benutzt man die Opcodes 192 bis 197. Um den Inhalt einer dieser Speicher auf den Calculatorstack zu legen, benutzt man die Opcodes 224 bis 229

Es gibt sogar Calculatoropcodes zum überspringen von Opcodes. Dazu gibt man hinter dem Befehlsopcode die Anzahl der zu überspringenden Opcodes an; diese kann auch negativ sein, um rückwärts zu springen.

Opcode: Ausführung:
51 unbedingter Sprung
0 Sprung nur dann, wenn das oberste Calculatorelement ungleich null ist.
53 ,b Springt nur, wenn das b-Register ungleich Null ist. Außerdem wird das b-Register dekrementiert.
Bitte sehen Sie sich dazu noch das Beispiel auf der nächsten Seite an!

```
a) rst 40                 Calculatoraufruf
b) Byt. #163              PI/2 auf Calculatorstack
c) Byt. #51               Sprung
   Byt. #3       ------,  3 Bytes sollen übersprun-
gen werden             |
d) Byt. #49            |  PI/2 duplizieren
e) Byt. #15            |  PI/2 + PI/2  (=PI)
f) Byt. #56      <----'   End of Calculator
g) ld a,2
   call 5633              Kanal #2 öffnen
h) call 11747             Oberstes Calculator-
element wird ausgedruckt.
i) ret
```

Es wird also nicht PI, sondern PI/2 (= "+STR$ (PI/2)+") ausgedruckt, da die beiden Opcodes zum Verdoppeln übersprungen worden sind.

So, das war die Erklärung aller nützlichen ROM-Routinen und des Calculators rst 40.

Anmerkung der Redaktion: Dieser Text entstammt dem Spectrum Programm „Die nuetzlichsten Routinen“ von Klaus Jahn.
http://www.worldofspectrum.org/infoseekid.cgi?id=0008636
Ich habe alle Texte in den PC übertragen und mit Umlaut-Zeichen versehen.

Retro Computer Match Teil 1

In dieser Serie will ich mal 8/16 Bit Rechner mit dem ZX Spectrum vergleichen. Wieso? Einfach so. Ich besitze einige Retro-Rechner, nicht zuletzt dank Wilko Schröter (Vielen Dank übrigens!).
Auch möchte ich ein Gedankenexperiment dazu machen: Was wären die Auswirkungen auf den Spectrum, wenn es diesen Rechner nie gegeben hätte?

Zum Einstand kommt ein ganz besonderer Rechner: Der **Texas Instruments TI99/4A**.
Der TI99/4A ist nach dem TI99/4 der zweite 16Bit Heimcomputer der je erschienen ist.

	ZX Spectrum 48	TI99/4A
Baujahr	1982	1981
CPU	Z80A 8 Bit	TM9900 16 Bit
Takt	3,5 MHz	3 MHz
RAM	48 KB	256 Byte
VideoRAM	16 KB (geteilt)	16 KB (dezidiert)
ROM	16 KB	28 KB
Grafikmodi	1	2 (3)
Textmodi	0	2 (1)
HW-Sprites	0	32 Mono (4/Zeile)
HW-Scroll	Nein	Nein
Auflösung	256x192	64x48+256x192
Farben	8 (15)	15 (+1 Transp.)
Sound	1 Kanal	3 Kanäle
Soundchip	ULA	TMS9919
OS	BASIC (MS)	BASIC (Eigen)
Medium	Kassette	Tape & GROM
Speech	Softwaresample	Modul-Super!
Tastatur	Folie und Gummi	Folie und Plastik

Man sieht schon, der Rechner war dem Spectrum theoretisch überlegen. Ein 16-Bit Prozessor kann einem 8-Bit nicht gefährlich werden. Zudem hatte der TM9900 eine Innovation: Es hatte nur einen echten Register in der CPU, jedoch zeigte dieses auf das „Scratchpad“ RAM welches mit maximaler Geschwindigkeit angesprochen wurde, also wie die ersten 256 Bytes beim C64. Der Vorteil war, dass man die dort gespeicherten restlichen Registerinhalte wesentlich schneller austauschen konnte als beim Z80 mit Stack. Auch die Grafik war dank TMS9918A/TMS9929A (wird auch in MSX verwendet) sehr gut. Attributgröße war 8x1 (In BASIC nicht verwendbar), oder im Char-Modus 8x8, wie beim Spectrum, wobei 3 Charsets mit je 256 Zeichen zur verfügung standen, also ließ sich die Hires-Chargrafik mit Spectrum vergleichen, bis darauf dass der Spectrum keine Hardwaresprites hat, der TI aber 32 Monochrome beherrscht (4 in einer Zeile). Auch hat der TMS Videochip eine Genlock-Funktion, die aber beim TI nicht benutzt wird. Ursprünglich sollte der TI99/4 ein 8 Bit Rechner werden, aber der Chip wies Fehler auf und konnte so nicht verwendet werden. Also baute man in die 8-Bit Hardware mit etwas Adaptierungsarbeit den 16-Bit Chip, was aber alle weiter unten genannten Nachteile nach sich zog.
Aber...
Texas Instruments machte aus Gewinnsucht (ist ja eine amerikanische Firma) ein paar entscheidende Fehler beim Design.
Man wollte nicht seine 100.000-de Dollar teure 16-Bit Workstation mit gleichem Prozessor überflüssig machen, also was tat TI-Vorstand? Er befahl den Designern den Rechner künstlich zu verlangsamen. Statt 10 MHz der Workstation, taktete man die CPU auf 3,3 MHz herunter, aber das war nicht genug, das 8-Bit RAM wurde belassen, statt es auf 16 Bit zu erweitern (übrigens gleiche RAM-Bausteine wie Spectrum Video-RAM – TM6116, man stellte sie ja selbst her). Da der Rechner aber nur einen dezidierten Video-RAM hatte auf dem der Videoprozessor Zugriff hatte, mußte der Prozessor jedes einzelne Byte von dort über einen 16-8 Bit Multiplexer holen. Weil BASIC-Programme im Video-RAM abgelegt wurden, waren sie viel langsamer als wenn sie im normalen RAM

wären, der direkt von der CPU adressiert werden kann. Erweitert man den Rechner auf 32 KB mit SRAM, lässt sich die Ablaufgeschwindigkeit von Extended BASIC (welches in Modulform kommt und als einziges Speichererweiterung unterstützt), verdoppeln. Zudem ist das BASIC doppelt interpretiert (GPL Interpreter belegt einen Großteil des ROM und übersetzt den in Pseudo-Code geschriebenen BASIC-Interpreten in TM9900 Assembly), wodurch die Geschwindigkeit weiter sinkt, und zusätzlich war die Rechengenauigkeit doppelt so hoch wie notwendig, dadurch auch langsamer. Als die Manager bemerkten dass das BASIC langsamer als beim Commodore VC20 war, war es schon zu spät. Mit einem vernünftig in MC geschriebenen Interpreter wäre der TI99/4A etwa 10 mal schneller als jetzt, und dadurch dem Mitbewerb überlegen. Der P-Code wurde wohl für den geplanten 8-Bit Chip entwickelt.

Eine weitere in der Reihe der Fehlentscheidungen war dass TI den ganzen Zubehörmarkt kontrollieren wollte. Aus der Überlegung heraus „Wenn niemand anderer Soft- und Hardware dafür anbieten kann, sind unsere Gewinne noch höher“. Ätsch! War gar nicht so. Zuerst einmal wurden statt EPROM nur GROM Module eingesetzt, die nur TI produzierte (GROM haben weniger Pins und konnten daher nicht direkt adressiert werden. Will man das 3000-ste Byte lesen, mußte man die 2999 Bytes davor auch auslesen). Es dürften nur Hersteller Spiele machen, die dem TI-Vorstand genehm waren. Atari hat z.B. keine Spiele für den TI rausgebracht. Alles was die Schnittstellen betraf, war streng geheim. Es gab keine Schaltpläne, keine Dokumentation, das schreckte viele Firmen und Hobbyprogrammierer (die nur in BASIC programmieren durften weil Maschinencode nur von Cartridge gestartet werden konnte) ab. Was aber eine zu geringe Softwarebasis bedeutet, kann man sich ausmalen. Mit dem immer noch interpretierten „Extended BASIC“ wurde das MC Problem umgangen,

Trotz der Tatsache dass TI alle Chips für TI-99/4A selbst produzierte und ihn daher sehr günstig herstellen konnte und trotz der beigen Low-Cost (QI=Quality Improved) Version ohne gebürstetes Alu-Blech konnte TI die Preisschlacht mit Commodore nicht gewinnen. Commodore optimierte die Produktion ständig um billiger zu sein (z.B. die Abschirmung aus Pappe), während TI immer darauf bestand hochwertig zu produzieren in der Hoffnung dass die Kunden das anerkennen werden. Letztendlich mussten sie sich aus dem Heimcomputermarkt zurückziehen weil TI die Geräte unter den Produktionskosten verkaufen musste (wir erinnern uns, Gewinne werden nur mit Zubehör gemacht) und schließlich monatlich Millionenverluste machte.

Wäre das BASIC besser und schneller gewesen (mit Möglichkeit Assembler Routinen zu starten, also wie „Extended BASIC“, nur eben schneller), mehr Speicher mit schnellerer Einbindung, sowie offene Dokumentation, wäre der Rechner nicht zu stoppen gewesen.

Nichtsdestotrotz hat das Gerät heute noch sehr viele Fans die sich immer wieder treffen. In Wien gibt es z.B. ein monatliches TI99/4A-Club-Treffen. Es existiert ein Compact Flash Interface für den TI.

Was wäre die Auswirkung auf den Spectrum wenn es denTI99/4A nie gegeben hätte?

Die Auswirkung wäre nur gering. Der Markt hätte sich auf die restlichen Konkurrenzprodukte, darunter den Spectrum, aufgeteilt, jedoch nur in Europa. In USA wäre Commodore weiterhin stark gewesen, jedoch nicht derart im Zugzwang die Preise zu senken, was Timex wiederum gestärkt hätte.

Trotz all dem Negativen ist und bleibt der TI99/4a ein sehr interessantes Retro-Gerät.

LCD

English summary

In this article I compare the 8 Bit Spectrum with a 16 Bit TI99/4a, which suffers from some design and marketing faults. There is also a TI99/4a club in Vienna:
http://ti99blog.webs.com/
LCD

Servicing Sinclair Computers Part 1

http://www.worldofspectrum.org/hardware/rep2.html

Mike Phelans article "The Lid off Microcomputers" last year evoked considerable interest among readers. It's clear that many of you are keen to get to grips with these devices which seem to be taking over the electronics world. They come in quite a lot of different types and forms. The single-chip types used in VCRs, washing machines and cars are special-purpose devices: they are not ideal subjects for an initial assault even if the wife would let you at them. So forget these and Head Office's IBM main-frame computer. In this series we're going to deal with some of Sir Clive's products, which happen to be among the cheapest, simplest and most abundant to have been put on the market. They have the added advantage that spares are readily available, which must appeal to anyone used to the problems of the TV servicing trade.

Servicing Equipment

One of the first thoughts that will probably cross your mind is the cost of the servicing equipment required. Perhaps, like me you've wandered round the exhibitions looking at £2000 plus scopes and sneaking envious glances at the logic analysts – "no self-respecting computer repair organisation should be without one". I often wonder whether I'd have the time to learn how to use one of these even if I could afford it. But fear not. The most I recommend, in addition to the usual TV servicing equipment (scope, multimeter, bench PSU etc.), is a logic probe. Even this could be made from one of the many circuits that have been published if you wish to keep the initial expenditure down.

In some ways a logic probe duplicates a scope, which can be used instead. Its very much quicker and easier however to use a device with an in-built indicator when checking for pulses along the pins of an ic.: you don't have to look up and check each response on the scope. The signals you'll be looking at are generally TTL ones (0-5V), so if you use a scope for the purpose its sensitivity need be no greater than that required for TV work, though the essential high-impedance probe does reduce the signal a bit. The crystal oscillator used in a microcomputer often runs at about 14Mhz, ie. rather higher than the basic frequencies encountered in a TV set, but there's rarely need to study waveforms at this frequency and the more critical system clock frequency is in the 2-4MHz range.

The Central Processing Unit

The system clock frequency just mentioned depends on the type of central processing unit (CPU, i.e. microprocessor) used in the microcomputer. Sinclair use the Z80 family: the Z80 runs at a clock frequency of 2MHz, the Z80A at 4MHz and the Z80B at 6MHz. The Z80A, which is used in all the microcomputers we'll be considering, is housed in a 40 pin package - Fig.1 shows the connections. Some of the names used may be new to you, so we'll briefly run through the pins and their functions.

Note that the address and data bus lines are the only "active high" ones: all the rest go low (to 0V) when operative and should thus be written as RESET for example (= not reset when high, i.e. 5V).

Designation	**Function**
5V, GND	Chips 5V supply and chassis connections.
A0-A15	Address bus outputs which are Tri-state, active high. i.e. either high (active)=5V, low = 0V, or open-circuit (high impedance). This later state allows other devices to control the line without loading problems. A0 is the least significant bit. These 16 lines can address 64k binary addresses (65,536 in decimal as 1k=1024 decimal, ie 64x1024)
D0-D7	Data bus lines (in/out), again tri-state active high. D0 is the

least significant bit. These eight lines carry the data to and from the CPU. They represent 1 byte in binary (256 decimal).

The CPU generates signals on the following six lines to inform and control the other devices in the system. When the lines carry signals from the rest of the system the CPU Output is tri-state.

Designation	**Function**
M1	Machine cycle one, active low. Used by the CPU to signal that a particular loading cycle is being carried out.
MREQ	Memory request, active low, tri-state. Indicates that the address bus holds a valid address for a memory read or write operation.
IORQ	Input/output request, active low, tri-state. Indicates that the lower half of the address bus holds an address for use by an input/output device.
RD	Memory read, active low, tri-state. Signals read to a memory or input/output device.
WR	Memory write, active low, tri-state. Signals write to a memory or input/output device.
RFSH	Refresh, active low. Indicates that memory refresh is taking place. Every address in a dynamic memory has to be refreshed at 2msec intervals. This line signals that a refresh is taking place. More on this when we come to memories.

The remaining lines are used by the system to initiate action or to indicate that action has been taken.

Designation	**Function**
HALT	Halt output, active low. CPU output signal indicating that it has obeyed a software halt instruction.
WAIT	Wait input, active low. Allows external devices to halt the CPU. Must be of short duration because refresh is stopped.
NMI	Non-Maskable interupt input, active low. Allows an external device to interupt the CPU and make it carry out a special software program.
INT	Interupt request input, active low. Similar to NMI but disregarded when instructed by the program running.
RESET	Reset input, active low. Resets the CPU to start address (000 Hex). Takes place automatically at switch on. Refresh stops and dynamic memory is cleared.
Clock	System clock input. This signal controls the speed of the system and the CPU's operations are synchronised to it. The frequency depends on the CPU but is usually 2-4 Mhz.
BUSREQ	Bus request input, active low. This line is activated by an external device when it requires control of the system. The CPU outputs other than BUSAK go to the open-circuit state and refresh ceases.
BUSAK	Bus acknowledge output, active low. Reply signal ro BUSRQ.

It’s not necessary to worry too much about the precise function of each of the above lines: the important thing is whether a high, low or a dynamic signal is to be expected. For instance the RESET line has to be high (except at switch on) or the CPU ceases to function. Many other lines, particularly the address and data lines, are continually changing even when the computer is apparently doing nothing. At all times the refresh system is putting out addresses and indicating their presence via the RFSH line. Although this dynamic action is rarely cyclic and often produces a meaningless pattern on an oscilloscope we shall see later that it can be

turned to advantage when fault finding.

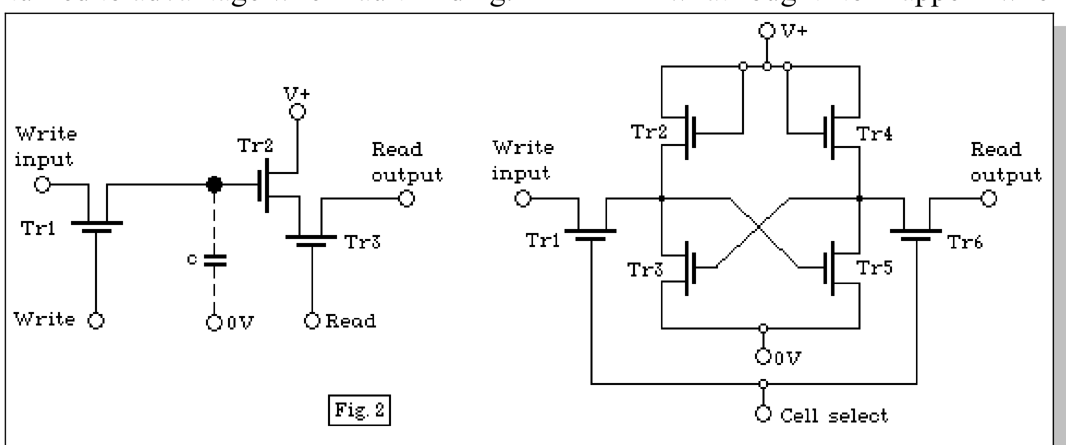


Fig2. Examples of basic MOS dynamic RAM (left) and static RAM (right) memory cells. In the DRAM cell the data bit is stored as an electrical charge by the capacitor shown as 'c', in practice this is Tr2's input capacitance. The SRAM cell uses a bistable circuit for storage (Tr3/5, with Tr2/4 as loads): The cell selection lines enables data to be written in or read out. The DRAM memory cell is much simpler, giving greater storage per square area of silicon chip, but requires refreshing every 2msec.

Programming Knowledge

After dismissing the need for a detailed knowledge of the operation of the CPU this is perhaps an appropriate time to stress the need for some knowledge of programming. Unlike other electronic systems the computer 's controlled by the software program and the program resident in the ROM. To expect to be able to carry out fault diagnosis without an understanding of how to program the machine is naive, and a knowledge of how the system is designed and functions can be a great help. Most home computer handbooks give instructions on programming - usually in the 'universal' BASIC (Beginners All-purpose Symbolic Instruction Code) language. Unfortunately however there are different BASIC systems. Sinclair BASIC is one of the most way-out, and although devotees swear by it the differences between it and others are considerable. So too are the computer start-up sequences, and it's a great help if one knows what ought to happen when the machine is switched on. To sum up, I'd not advise anyone to attempt repairing a machine before obtaining practice and experience of its operation. More on this when we come to the ZX81 next month.

Memory Chips

While discussing the CPU we mentioned dynamic memories. We'd better next look at the memory family. As you probably know by now there are two basic types of memory, ROM (read only memory) and RAM (random access memory). The former are preprogrammed and non-volatile, i.e. they contain the, manufacturer's program for operating the machine/system: this program is permanent and unchangeable. The memory in a RAM can be changed however and is used to store the data produced by the computer system. Its volatile in that all the data is lost if the supply voltage fails, but some *RAMs* are even more vulnerable. These are the dynamic memories (DRAMs) that require constant refreshing. With these each binary bit of every data word is stored as an electrical charge at the input of a high input-impedance transistor. This charge leaks away in about 2msec. So each bit - and there may be as many as 64K bits - has to be recharged every 2msec. A daunting task you may think.

Its not quite as bad as it looks however since another feature of the dynamic RAM helps. We said earlier that sixteen address lines are required to count up to 64K, So a DRAM is going to be a multi-pin device with probably over twenty pins. In fact it has sixteen. This is achieved by matrixing the address lines. The address matrix is formed of rows and columns and a system built into the chip switches eight of

the pins first to one and then the other. Externally the sixteen address lines are also switched, first to one set of eight addresses then to the remaining eight. Both these switching operations are synchronised by the CPU: the data appears on the data bus at a precise time and is read by the CPU or the memory in sync with the switching.

Back to our refresh problem. The internal circuitry that does the row/column switching also assists with the refresh by recharging all the column bits each time a row is addressed. It's necessary to refresh only the rows therefore to refresh the memory completely. The number of refresh operations is thus reduced from 64K to ~ than 256.

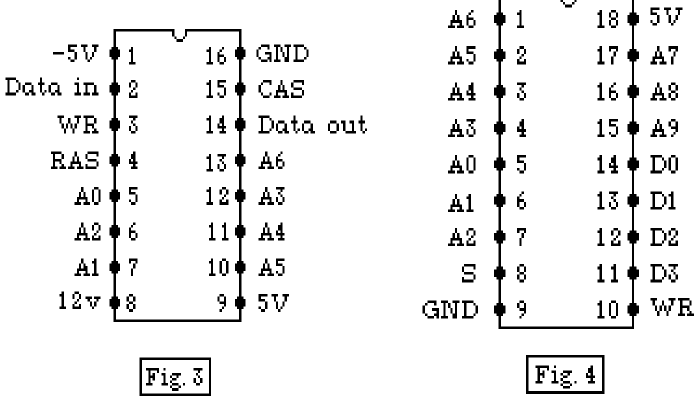


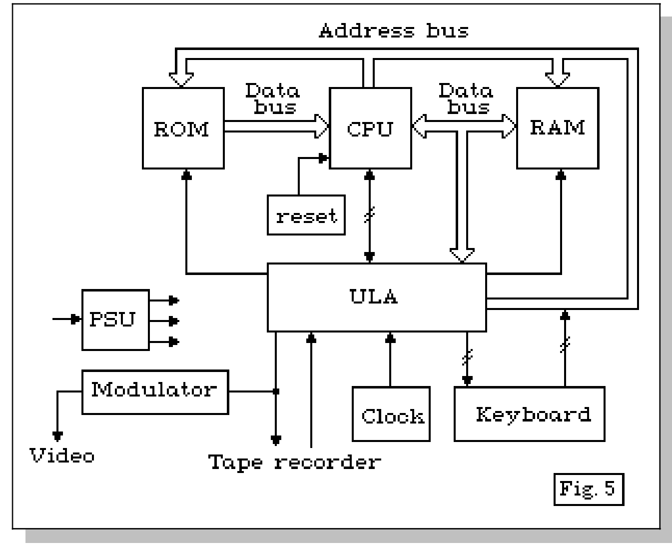
Fig. 3 shows the pin connections for the 4116 DRAM. The RAS (row address strobe) and CAS (column address strobe) pins are used to synchronise the switching. This is a 16K by one bit (serial bits enter at pin 2 and leave at pin 14) memory. Despite requiring three voltage supplies (12V at pin 8, 5v at pin 9 and -5V at pin 1) it's still only a 16-pin device. Seven lines (A0-A6) are used for the addresses. The other pins are chassis (16) and WR (3) – read/write select. Compare this with the 2114 static RAM - see Fig. 4. Static RAMs use flip-flops to store the bits. This device has only one supply and has a 1K by four bit memory, i.e. it can store 1,024 four-bit number. Its storage capacity is a quarter of that of the 4116, yet 18 pins are required.

The Z80 is an eight-bit microprocessor, these eight bits being known as one byte. It connects with an eight-line data bus and when it addresses a memory location it expects to receive an eight-bit instruction to direct its operation. The memory chips are arranged to meet this requirement: in the case of the 4116 eight of these devices would be used, providing a 16Kbyte memory; as the 2114 is a four-bit memory two of these would provide a 1Kbyte memory.

When memories are assembled in this way the address lines are common to all the memory chips. If there **is** more than one memory group with the same address pin confusion would exist if they all unloaded their data together. To overcome this problem memories are generally provided with chip select (CS or S) or chip enable (CE) pins often more than one. These switch the memory on only when they are active.

Suppose for example that four 2114 memories are paralleled on the address bus to provide 2K x 8 bits of memory. If an address in the 0-1K range is required the first two memories will be enabled whereas for an address in the 1-2K range the decoder detects the higher number (usually the next binary address line) and enables only the second memory pair. The data pins are tri-state and remain in the high-impedance state until the memory is enabled.

Complete Microcomputer System



It's time to look at the computer system as a whole. Fig. **5** shows a block diagram for a simple computer using the absolute minimum of components. We've already dealt with most of

the important items - in fact the only major item remaining is the ULA (Uncommitted Logic Army). This is a collection of logic circuits assembled to the computer manufacturer's specification, replacing the many standard and special logic chips that would otherwise be required. In this example it would provide the address decoding, produce the clock pulses, decode the keyboard, condition the analogue tape input signals and control the output of data to the tape recorder and video modulator. Some of these tasks are carried out by the ULA alone, some are controlled by the CPU.

The clock oscillator for example is driven and divided automatically and the chip select signals are continually produced by the ULA logic alone. The output to the video modulator however is a complex signal consisting of screen character pixels and sync signals: the CPU is needed to sequence these correctly. The CPU and ULA also combine to carry out the keyboard decoding. Electrically the keyboard is organised in matrix form, one set of contact connections running from top to bottom and the other along the rows of keys. Signals are sequenced in the vertical lines via the ULA, a key producing a pulse on the address line. This pulse can then be decoded in a time sequence to find the key pressed.

We've now covered the major components in this simple example of a microcomputer. But we haven't finished with Fig. 5 because its not a fictitious example. In fact it's a block diagram of the ZX81, the simplest Sinclair computer produced. Because of its many limitations this model has become obsolete and has little marketable value, As a result ZX81's can be purchased second-hand for as little as £10 They make ideal initial practice machines. Anyone contemplating microcomputer servicing would, if he doesn't already have a machine, be well advised to buy one of these for both the software and hardware experience.

Next month well look at the ZX81 in detail and start to establish a fault-finding procedure.

Deutsche Zusammenfassung

Hier wird erklärt, bevor es an detaillierte Reparaturanleitungen geht, wie die einzelnen Komponenten im Spectrum funktionieren. Ein besonderes Augenmerk wird dabei auf die CPU und RAM gelegt. Alle Signalleitungen der CPU werden halbwegs detailliert erklärt, sowie der Unterschied zwischen DRAM und SRAM. Auch ein Blockdiagramm (ZX81) wird hier gezeigt um zu erklären wie ein Computer funktioniert.

Adventurelösung: Das Kreuz des Königs

Hallo Adventurefreunde!!

In unserer Reihe mit Adventurelösungen sind wir wieder einmal bei einem deutschsprachigen Vertreter dieser Programmart angelangt. Wir befassen uns heute mit dem Programm "Das Kreuz des Königs" von Peter Seyl aus dem Jahre 1985. Wie damals üblich, wurde auch dieses Adventure vollkommen in BASIC geschrieben und stellt mit seinen 34 Locations einen Vertreter der Adventures dar, die man relativ leicht lösen kann. Sowohl der Umfang als auch der Schwierigkeitsgrad sind nicht sehr hoch, das soll aber nicht heißen, dass dieses Programm nicht einer näheren Betrachtung wert wäre. Kommen wir kurz zur Hintergrundgeschichte, die den Rahmen zu diesem Programm bildet. In diesem Programm sind wir in einer Burg älteren Datums angekommen und unser primäres Ziel ist es, mit allen Schätzen aus der Burg zu verschwinden. Dabei wird sich herausstellen, dass bestimmte Teile in diesem Programm mittels Zufallsgenerator gesteuert werden, was manchmal unser Fortkommen erschwert. Die Locations, wie aus beiliegendem Plan ersichtlich, sind ziemlich logisch und übersichtlich aufgebaut. Im Einzelnen stoßen wir auf unseren Erkundungen auf folgende Räumlichkeiten:

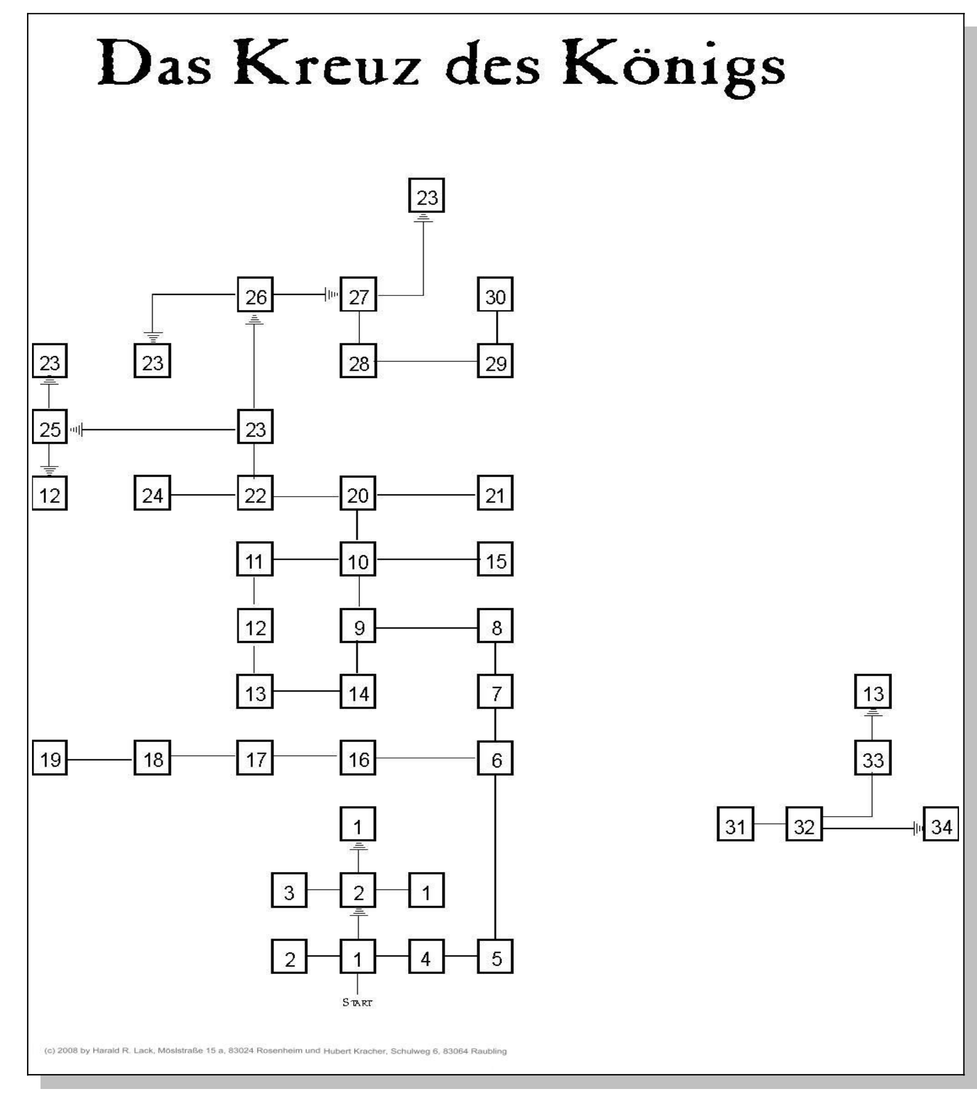
01) In der Eingangshalle

02) Im Wachzimmer
03) In der Waffenkammer / Schwert
04) Im Treppenhaus A
05) Im Eckturm A
06) Im Treppenhaus B
07) In der Halle / Zange
08) Im Eckturm B
09) Im Hof
10) Im Speisezimmer / Brot
11) Im Eckturm C:
12) Im Treppenhaus C / Lampe
13) Im Gerümpelkeller / Schlüssel, Münze
14) Im Weinkeller / Wein
15) In der Küche / Ring
16) Im Wachraum
17) Im Schlafraum / Spiegel
18) Im Lagerraum
19) Im Turmzimmer / Dietrich
20) In einem Speisezimmer
21) In der Folterkammer / Dolch
22) In einem schönen blühenden Garten
23) Im Vorhof des großen Turmes
24) In einer Fallgrube
25) In einem Nord/Süd Gang
26) In einem hohen Turm
27) In einem engen dunklen Hof / Pfeil
28) In der königlichen Stahlkammer / Goldbarren
29) Im Löwenkäfig
30) In der königlichen Schatzkammer / Silberkreuz
31) Im Gefängnis
32) Im Gang vor dem Kerker
33) In einem engen niedrigen Nord/Süd Gang
34) Im Wachraum

Soviel zu den Locations und Gegenständen, die wir so antreffen.

Bevor wir jetzt ins Geschehen einsteigen, noch eine Anmerkung zur Bedienung. Naturgemäß benötigt ein BASIC Programm immer ein bisschen länger mit der Abarbeitung der Eingaben oder damit, neue Befehle entgegen zu nehmen. Etwas Beherrschung kann bei diesem Programm also nicht schaden. Mit dem Befehl LIST kann man nachsehen, was man so alles an Gepäck mit sich schleppt. Durch die Eingabe von "?" oder Punkte, erfährt man den momentanen Punktestand und weiß damit, wie erfolgreich man bisher war. Mit Schau kann man sich die aktuelle Location neu betrachten. Doch das sind sicherlich Dinge, die man als bekannt voraus setzen kann. Von Zeit zu Zeit kann es passieren, dass uns patrouillierende Wachen (bis zu drei Stück) entdecken oder sogar anhalten. Es ist daher oft ratsam, das Weite zu suchen. Mit dem Befehl FLIEH können wir ja mal unser Glück versuchen. Oft klappt es, manchmal landen wir aber auch im Gefängnis. Wenn uns nur eine Wache gegenübersteht und man entsprechend bewaffnet ist, gelingt es mitunter auch den Typen in die Flucht zu schlagen. Durch den Befehl BEKÄMPFE WACHMANN kann man ja mal sehen was dabei rauskommt. Sollten wir jedoch, wie in den meisten Fällen im Gefängnis landen, so bitte keine Panik oder gar Verzweiflung aufkommen lassen. Sollten wir schon im Besitz des Schlüssels sein, so können wir uns selbst befreien, da dieser die Tür in der östlichen Mauer sperrt. Sollte uns das Schicksal jedoch schon relativ früh ereilen und wir noch keinen Schlüssel besitzen, so ist Beharrlichkeit ein Weg zu Erfolg. Im Gefängnis geben wir dann solange den Befehl WARTE ein, bis wir ein Knacken am Schloß der Türe wahrnehmen (meist nach dem 4. Versuch). Wie es scheint, haben wir Sympathisanten im Schloss, die uns befreien. Doch nun sind der Worte genug gesprochen, lasst Taten folgen ....

Wir stehen an unserem Ausgangsort (in der Eingangshalle) und schauen uns um. Nachdem es in alle vier Himmelsrichtungen geht, können wir uns mal eine aussuchen. Sollten wir uns in südlicher Richtung davon machen, so erfahren wir, dass es hier nach draußen geht und wir offensichtlich auf der Flucht sind und schon aufgeben wollen. Das können wir natürlich nicht auf uns sitzen lassen. Deshalb stürmen wir nach Vorne (sprich Norden) und kommen ins Wachzimmer. Mit etwas Glück bemerkt uns keiner und wir gehen von hier aus weiter nach Westen, wo wir in die Waffenkammer gelangen. Dort gibt es ein Schwert, das wir mitnehmen. So bewaffnet fühlen wir uns gleich besser. Also



geht es jetzt wieder zurück nach Osten (der einzige Ausweg) und mit viel Glück ist im Wachzimmer noch Ruhe, so dass wir unbehelligt bleiben. Von hier aus geht es nach Norden und Osten und egal welche der Richtungen wir einschlagen, wir landen wieder in der Eingangshalle. Na, der zuständige Architekt, der diese Burg plante, scheint auch etwas verwirrt gewesen zu sein. Jedenfalls sind wir jetzt wieder da, wo wir schon einmal waren. Also machen wir uns weiter nach Osten und gelangen in ein Treppenhaus und von dort aus weiter in einen Eckturm (Location 6). Von hier aus geht es nun weiter nach Norden und wir kommen wieder in ein Treppenhaus. Wenn wir den Text, der uns nun angezeigt wird genaues lesen, erfahren wir, daß in der Westwand eine Tür ist. Das riecht nach Abenteuer und deshalb gehen wir, sogleich in diese Richtung. Wir gelangen in Raum 16 (Wachraum) und hier werden wir dann mal wieder versuchen, keine Probleme mit den Wachen zu haben. Weiter geht es nach Westen und wir sind jetzt im Schlafraum, in dem wir einen Spiegel sehen können. Wenn wir ihn aufnehmen und drehen, so sehen wir auf seiner Rückseite ein Auge. Dieses Symbol werden wir später noch antreffen. Also wieder weg mit dem Spiegel und weiter nach Westen in den Lagerraum. Hier scheint es außer alten Sachen nichts zu geben was man eventuell brauchen könnte und so gehen wir nochmals nach Westen und gelangen ins Turmzimmer. An der Nordwand ist ein offener Schrank zu sehen, in dem ein Dietrich hängt. Da kann man nur sagen - her damit. So ausgestattet, machen wir uns auf den Weg zurück zur Location 6. Wir gehen in nördlicher Richtung über die Halle (Location 7) bis zum Eckturm (Location 8) und wenden uns gen Westen. Wir gelangen in einen Hof, in dem der Dreck meterhoch liegt. Also gehen wir nach Norden und kommen ins Speisezimmer. Dort gibt es Brot, das wir mitnehmen könnten, damit wir was zu beißen haben. Aber wir sollten uns nicht unnötig damit aufhalten. Besser ist es, wir gehen von hier aus nach Westen und kommen über den Eckturm (Location 11) jetzt in südlicher Richtung in .... richtig ein Treppenhaus. Dieses unterscheidet sich von den anderen dadurch, daß hier ein recht nützlicher Gegenstand zu finden

ist, nämlich eine Lampe. Wir nehmen sie an uns und gehen so ausgerüstet weiter nach Süden. Wir sind im Gerümpelkeller angekommen. Dadurch, dass wir die Lampe dabei haben, können wir hier nun unter anderem einen Schlüssel ausmachen. Den nehmen wir mit. Sollten wir in der Zwischenzeit schon mal im Gefängnis gelandet sein (Location 31) so kommen wir von dort aus über den östlichen Weg in den Gang vor den Kerker. Von hier aus können wir nach oben durch eine Öffnung und kommen in den engen niedrigen Nord/Süd Gang (Location 33) und von hier aus weiter nach Norden (in Location 13). Wir können also auch über diesen Weg an den Schlüssel gelangen (vorher aber Lampe aufnehmen). Falls wir den Dietrich noch nicht haben, müsen wir ihn uns noch schnell holen. Egal über welchen Weg wir bis hierher gelangt sind, jetzt ist es an der Zeit sich wieder im Speisezimmer (Location 10) einzufinden. Den Weinkeller, den wir eventuell auf unserem Weg passiert haben, brauchen wir keine große Bedeutung zu schenken. Zwar finden wir dort, wie es sich gehört. Wein vor, aber wenn wir ihn trinken sind wir sehr bald hakendicht. Also sollten wir uns beherrschen. Nun aber weiter im Text. Östlich vom Speisezimmer ist die Küche, die wir nun aufsuchen. Hier sitzt ein Typ rum, der einem Koch recht ähnlich sieht und löffelt aus einer Schüssel. Da wir mit dem Schwert gut bewaffnet sind und auch sonst überschüssige Kraft haben, legen wir uns mit dem Koch an und bekämpfen ihn. Wenn uns das Glück hold ist, werden wir ihn töten und um seinen Hals ein Kettchen mit einem Ring vorfinden. Den Ring nehmen wir an uns. Dann aber nichts wie zurück zum Speisezimmer und den Ring gedreht. Wie von Geisterhand öffnet sich eine Geheimtüre in der Nordwand und ehe sie sich schließt, schlüpfen wir hindurch. Wir sind jetzt in einem weiteren Speisezimmer (Location 20) und von hier aus gehen wir nach Osten. Dies ist die Folterkammer. Hier es jede Menge "schöne" Folterinstrumente zu bestaunen. Wir können daran keinen Gefallen finden und nehmen uns als Andenken nur den herumliegenden Dolch mit. Dann aber gleich wieder zurück ins Speisezimmer und weiter nach Westen. Wir stehen in einem schönen blühenden Garten aus dem mehrere Wege herausführen. Sollten wir uns nach Westen orientieren, so landen wir in einer Fallgrube. Hier kommen wir nur noch heraus, wenn wir die Wachen rufen. Daß wir dann wieder im Gefängnis landen braucht wohl nicht erwähnt zu werden. Also gehen wir nach Norden und kommen in den Vorhof des großen Turmes (Location 23). Hier ist deutlich sichtbar ein Hebel angebracht, der sich in zwei Richtungen drehen lässt. Wenn wir ihn rechtsrum drehen, so öffnet sich eine Falltür und unsere ganze Anstrengung war umsonst - das Adventure ist aus. Also drehen wir mathematisch positiv linksrum und die Tür im Westen öffnet sich. Wenn wir nun diesen Weg einschlagen, so kommen wir in einen Nord/Süd Gang und von hier aus entweder in Location 12 oder wieder zurück in Location 23. Da dies zugegebenerweise...auch nicht sehr produktiv ist entscheiden wir uns also für Norden und kommen in einen hohen Turm (Location 26). Von hier aus geht es nur nach Osten oder nach unten und da wir nun schon mal hier sind gehen wir nach Osten. Wir kommen in einen engen dunklen Hof, in dem ein Pfeil herumliegt. Wir nehmen ihn mal mit. Wie wir erfahren ist im Süden eine Tür mit einem Auge darauf. Das Symbol kennen wir ja schon vom Spiegel her. Die Tür ist jedoch verschlossen. Deshalb kommt jetzt der mitgebrachte Dietrich zum Einsatz und wir können nach Süden gehen. Dies ist die königliche Stahlkammer und hier liegt ein Goldbarren. Da Gold ziemlich schwer ist, lassen wir den Barren liegen und lauschen dem Geräusch, das von Osten her an unser Ohr dringt. Wir benutzen den Schlüssel und können nach Osten gehen. Wir sind **im** Löwenkäfig und wenn wir Glück haben, frisst uns der Löwe nicht sofort auf. Todesmutig stellen wir uns der Gefahr und bekämpfen den Löwen. Mit dem entsprechenden Geschick können wir ihn schließlich doch besiegen und das Gefühl sagt uns, dass wir unserem Ziel ganz nahe sind. Und richtig! Wir gehen nach Norden und kommen in die königliche Schatzkammer. Dort liegt das sagenumwobene Silberkreuz, der wertvollste

aller Gegenstände. Natürlich lassen wir uns nicht zweimal bitten und nahmen das Kreuz mit. Unser Weg führt uns nun zurück bis zur Eingangshalle und von da aus können wir nun das Schloss verlassen. Das Kreuz des Königs ist gelöst.

Wie ihr seht, war es per Saldo gar nicht so schwierig. Ein wenig kompliziert machen es eigentlich nur die Zufallselemente und kleinen versteckten Gemeinheiten. Aber zum Glück lässt sich das alles mit etwas Geduld überbrücken. Soviel dazu von unserer Seite. Bis zum nächsten mal!

© 2008 by Harald R. Lack, Möslstraße 15 a, 83024 Rosenheim und Hubert Kracher, Schulweg 6, 83064 Raubling

English Summary

Hello adventurers,

yet another little German adventure program which is called "Das Kreuz des Königs" (the king's cross) published back in 1985. As many adventures at this time it is totally written using Basic language and with 34 locations to visit it is not too big. But this doesn't mean it is not worth to have a closer look at. As you can imagine, it is not very difficult to solve but nevertheless a nice little game to play. Sadly for our foreign readers you have to understand the german language to play this game. But what is the story behind this game. The player is faced with an old castle somewhere and our only wish is to find all the treasures hidden inside and take them with us. What makes the things a little difficult, is, some activities in this game are random and so sometime it is hard to carry on. If you should be in trouble with the game, we have put our solution and map in the magazine for your help, if needed.

Spectrum als Antrieb im Modellbau



Im Hobby-Eisenbahnbereich kann man sehr schön Seilbahnen einsetzen. Es gibt genug Beispiele. Probleme stellen meist die Endpunkte, evtl. Zwischenhalte (d.h. Ab- oder Umschaltung) dar. Hier setze ich den Spectrum – verbunden mit einem Schrittmotor – ein. Durch Ausprobieren kann man die Lauflängen exakt ermitteln und wiederholen. Schaltungen für die Ansteuerung gibt es viele im Internet. Ich benutze den L297 und 298; hierfür gibt es Leerplatinen, Bausätze und Fertigprodukte, wichtig war mir, dass die Clockregulierung mittels Taster erfolgen konnte.
Der Spectrum wird mittels 8255 zum IN/OUT-Regulator, wobei ich nur 2 von 24 Anschlüssen benutze.

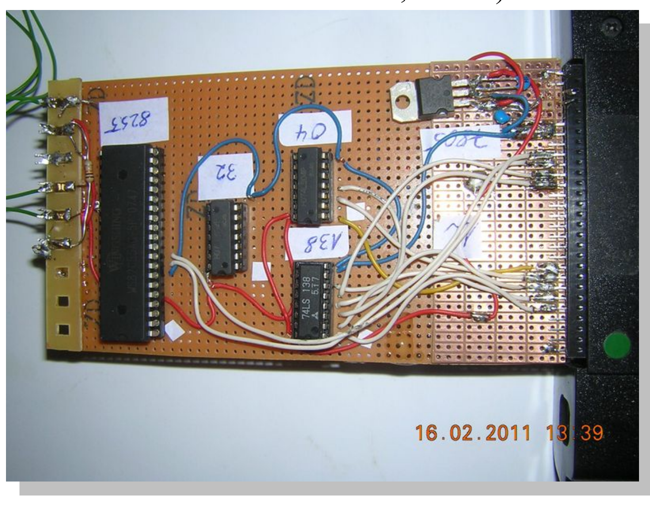


Ich habe dieses Interface nach Angaben des Buches Z80 Anwendungen gebaut.

Das Listing ist denkbar einfach und kann

jederzeit je nach Ansprüchen modifiziert werden.

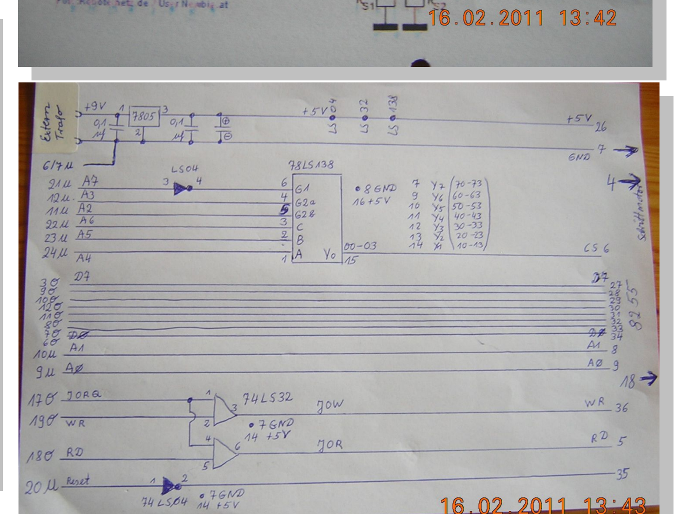
OUT 127,128 Initialisierung
OUT 3,128 Initialisierung von 8255 als nur Ausgabe
OUT 0,0 vorwärts oder rückwärts
OUT 0,1 rückwärts oder vorwärts
OUT 1,1 Augenblickliche Stellung des Motors
OUT 1,0 Weiterbewegung um Y Grad (bei mir waren es 1,8 Grad)



z.B.
10 OUT 127,128
20 OUT 3,128
30 FOR i = 1 TO 200
40 OUT 0,0
50 OUT 1,1
60 OUT 1,0
70 PAUSE x (bei mir nicht nötig gewesen)
80 NEXT i
ergibt eine Umdrehung bei mir

gleiches Listing, aber
40 OUT 0,1
ergibt bei mir 1 Umdrehung umgekehrt zu vorher

gleiches Listing ohne 30 und
70 GOTO 30
ergibt dauerhaftes Drehen vorwärts/rückwärts usw.
Die Anzahl der Impulse bestimmt also die Lauflänge des Seilbahn oder Bergbahn usw.



Viel Spaß
Harald Ködding (PUTITLE@aol.com)

Literatur:
Z80 Anwendungen von James W. Caffram, Sybeck 1984
www.roboternetz.de/wissen/index.php/Schrittmotoren, hieraus habe ich die Schaltzeichnung vom L297/L298 entnommen.

Speccy games after 1993, Part 10

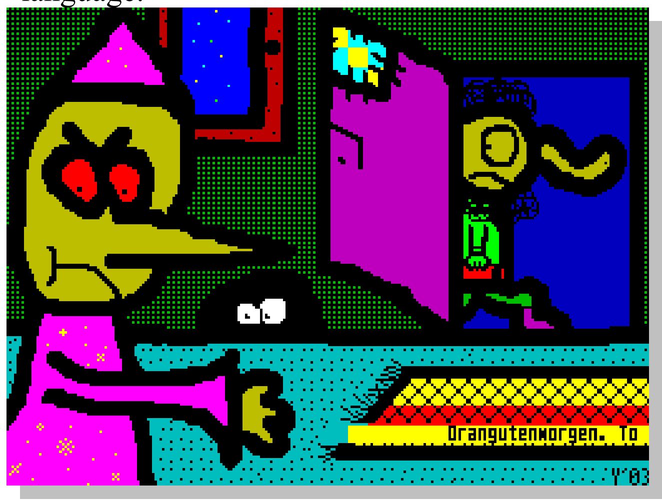
Poland

Hooy Program (or also known as H-PRG) is originally a Polish, actually an international demo group founded back in 1990 by Yerzmyey. Further members: Expirt, Gasman, Habib, Hellboj, TDM and Vavrzon. Ex-members: Adam, Mr. Hangman and Smarkoo.
The group members are very proud of the consistent quality of their works: they made crap demos in the past, and they do the same

nowadays. Both the audience and professionals regard their demos as the crapest ones. The team is active on the PC-category beside ZX Spectrum.

The title of their ZX magazine is Hooy Mag, it was survived two issues in 1998 and 2002.

At the 1993 turning point, released their quite primitive Smok text adventure in Polish language.



Droga du duplandu: forget it!

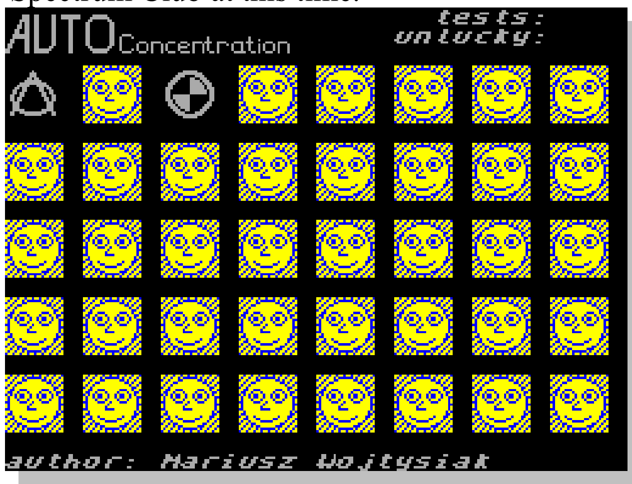
Their next production is an illustrated text adventure. Droda do duplandu is a crappy game again and (fortunately?) Polish language only - but English translation is promised. Was published in 2003, but Yerzmyey wrote its text between 1994-1997. It has C64 and Atari 8 bit version too.

The similar quality Ucieczka ze spejs-szipu (2003/2004) only released for C64 and Spectrum. From the title, easy to guess its language...

Mariusz Wojtysiak Polish programmer was infected with a Timex 2048 clone at the primary school age. After being a player, soon began to programming. Basic followed by Assembler in a short time, and soon became the member of Illusion group with Maniu nickname. Then changed to PC, and after graduating at computer sciences he is a programmer of IN-Software Poland at the moment.

In 1994 released his Auto Concentration memory game, which is essentially the same as the soon introduced game of WSS Team titled Chage It!, but here we can play with badges of cars.

The game was published in Scene+ disk/tape magazine, which was edited by the Austrian Spectrum Club at this time.



Auto Concentration: nice title

We can meet the name of Rafal Miazga first on the Spectrum Game Compo 2005. He was take part with the Pamela the Zombie Hunter game, in which we control a teenage girl a graveyard full of zombies. The next challenge was in the 2007 year with the Holidays in Amsterdam Frogger clone on the actual Mini Game Compo. After he changed his style, and give public the Chick&Brick titled Arcanoid clone. After completion a level with destroying all bricks, our reward is a picture of a nice girl (chick).



Mockatetris: another variant

Similar is the Mockatetris from 2008. Here, at the end of the levels, we can see handdrawn pictures from the author from games, which never released on Spectrum. And the classic

Tetris-shapes also got some new variants.
Skyscraper of Doom is again a sharp turn in style. This 2009-year arcade-adventure is playing in the future, and essentially maybe can be the analogue of the Dizzy-series in gamig style terms - of course with totally different graphics.

Weblinks:
Hooy Program: http://hooyprogram.republika.pl/
Mariusz Wojtysiak: http://wojtysiak.w.interia.pl/
Spectrum User Club: http://www.sintech-shop.de/home/spectrum-user-club.htm

Hungary

WSS Team (*Weird Science Software*) is primarily was well known by their demos in the 90's. Their first game came from 1995. This memory game is *Change It*! and the joint work of *Edy* and *Roger* (Edy: graphics, Roger machine code, Basic part: together).
After a long break in 2001 WSS reformed and came out with *Flash Beer* trilogy in 2003 (by Edy and *Pgyuri*). This game is originally a C64 classic - *Flasch Bier* and it is a Boulder Dash clone. Has a special loader.



Ghost Castles: my favourite from WSS

Next year's release was *TV Game*. It recalls the game consoles of 70's with some memorable unique features. One of them is the 288x192 pixels game area (using the border) and another is the multicolor mode. It offers 4 different games including tennis, wall-tennis, football and hockey.
The aim of *Ghost Castles* game (2005) is collecting nasty ghosts - no matter if it is done by alone, co-operative with your friend or with the Speccy - in that case you will do not need friends anymore:).
The curiosity of this game of supporting the *G-Tech interface* and *ROM-extension*.
G-Tech is the development of *Csaba Gondos*, the main constructor of Sinclair.hu. You can connect and use analogue joystick through a Kempston interface.
ROM-extension is a 256K ROM that gives 192K extras for the game (longer gameplay, digitised speech etc.).
Sadly, both devices are only used by this game. But hopefully soon will be supported in another games and emulators, as all documentations are free to download.
Freshest product is *Wizard of Wor*, which is a classical arcade game conversion. In the 80's was published on C64, *Atari 800 and 2600* as well for the *Bally Astrocade* console. After more than two decades, the Spectrum version also arrived.



Wizard of Wor: strange loader...

Good news from Edy: all games are free, and the followings will be free too. Every games can be bought on real tapes. Price of cassettes are covering the producing costs only, so sold without profit.

Weblink:
WSS Team: http://wss.sinclair.hu/

Will be continued   Richard Tarjan